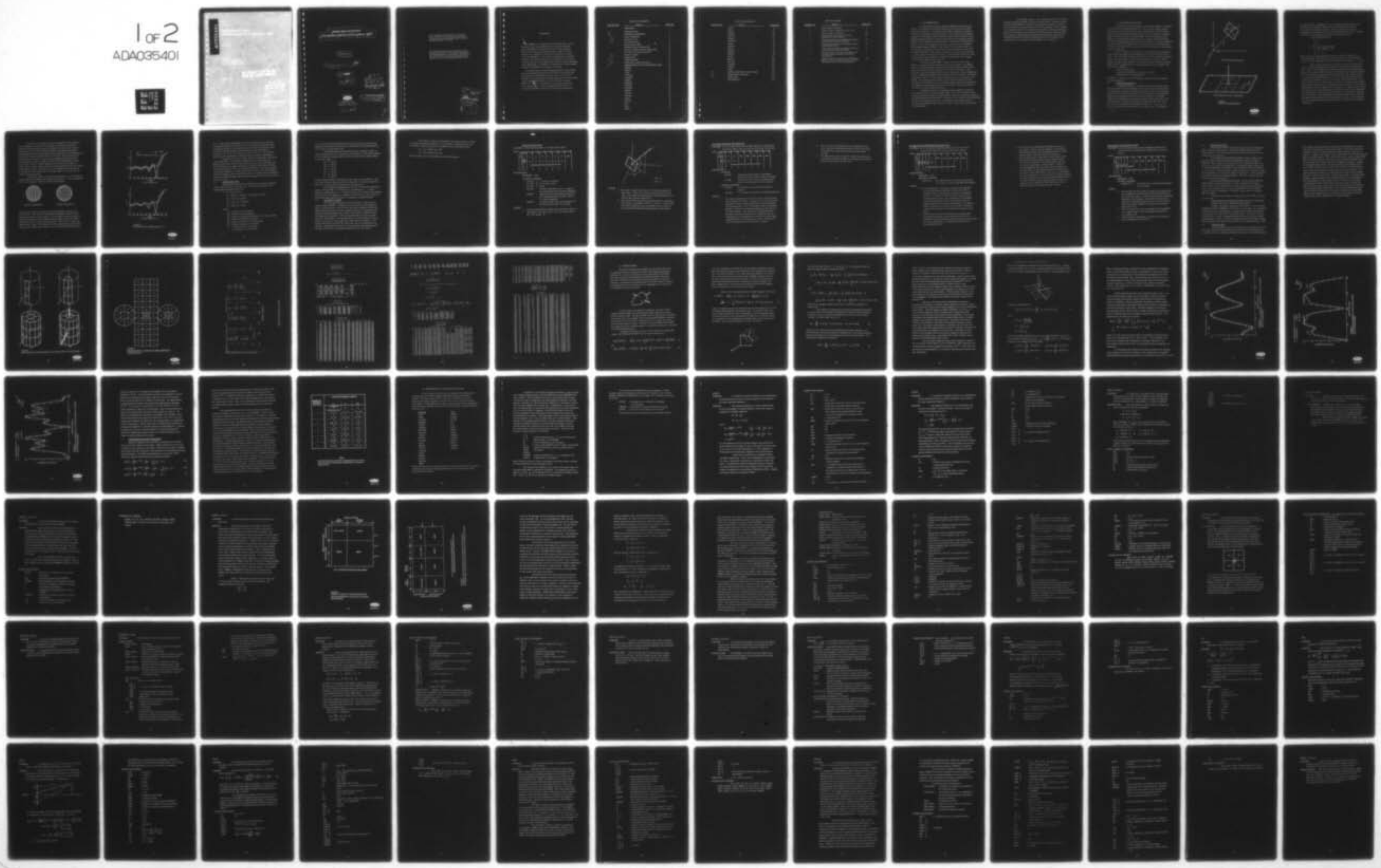
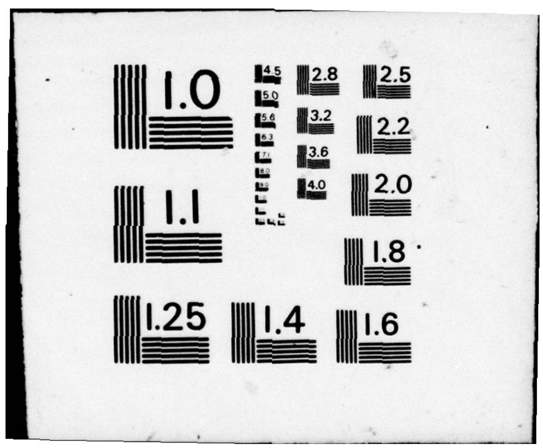
AD-A035 401 MB ASSOCIATES SAN RAMON CALIF F/G 9/5
ANTENNA MODELING PROGRAM SUPPLEMENTARY COMPUTER PROGRAM MANUAL --ETC(U)
JAN 75 N00014-71-C-0187
UNCLASSIFIED MB-R-75/4 NL

1 OF 2
ADA035401




1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

[illegible] MODELING PROGRAM

SUPPLEMENTARY COMPUTER PROGRAM [illegible] (AMP2)

[illegible]

Prepared by:
[illegible]
[illegible]
[illegible] No. 11

[illegible]

[illegible]

6

ANTENNA MODELING PROGRAM
SUPPLEMENTARY COMPUTER PROGRAM MANUAL (AMP2).

14 MB-R-75/4

15 N00014-72-C-0187

11 Jan[redacted]75

12 184p.

A005673

A031296

Prepared by
MBA
MBAssociates
Bollinger Canyon Road
San Ramon, California 94583

222 400

mt

This program was sponsored by the Army, Navy, and Air Force and was monitored by the Office of Naval Research under Contract Number N00014-71-C-0187.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies of any of the sponsoring agencies or of the United States Government.

FOREWORD

This manual is a supplement to the Engineering, User's, and Systems manuals prepared for the Antenna Modeling Program (AMP). This document describes the operation, theory, and coding of the changes made to AMP in order to decrease the running time for large voluminous structures with wire appendages. The options incorporated are surface modeling with surface patches as an alternative to wire grid modeling, and the use of approximate structure matrix elements where appropriate. In addition, an option for the precautionary dumping of temporary file storage is included.

The AMP code as modified (AMP2) has been implemented on the Naval Ship Engineering Center CDC 6700 and has been delivered to the Naval Research Laboratory, U.S. Army Strategic Communications Command, and the Rome Air Development Center, under the Office of Naval Research Contract N00014-71-C-0187. The program is under the direction of E. S. Selden and G. J. Burke of MBAssociates and M. L. Musselman and R. K. Royce of Naval Research Laboratory.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

The Hybrid Wire-Surface Program (AMP2) described in this manual is an extension of the basic Antenna Modeling Program (AMP) detailed in references 1, 2, and 3 with the added capability of modeling surfaces of voluminous conducting bodies via the magnetic field integral equation. The program can be used with wires alone in the same way as AMP employing the thin wire form of the electric field integral equation (EFIE). When modeling voluminous bodies, surfaces are represented by patches on which the surface currents are computed from the magnetic field integral equation (MFIE). Any number of wires and surfaces may be included in a model with mutual interactions computed by a hybrid of the MFIE and EFIE. Wires may be connected to surfaces at the centers of surface patches although not at edges. The program may also be used with surfaces alone, but only for scattering calculations since voltage sources may be applied only on wires.

The basic program AMP, using the thin wire form of the EFIE, can be used to model surfaces by use of wire grids. The MFIE, however, has been found to yield more accurate results with less computation time than wire grid modeling for voluminous structures. Also fewer decisions are required of the user in modeling a surface with patches than in choosing the locations and directions of wires in a grid. There are limitations, however, in the types of surfaces that can be modeled by the MFIE. The surface must be closed, such as that representing a thick solid body. Thin bodies, such as conducting plates cannot be modeled with the MFIE since two parallel surfaces close together (the front and back) will result in severe numerical errors.

In addition to surface modeling, program AMP2 provides two other options not in the basic program AMP. The time for matrix filling can be reduced by the use of an approximation when the interaction distances are greater than a specified value. In addition a time interval can be specified at which intermediate results will be dumped to file storage to permit restarting the program after a machine failure. Execution continues after each such dump.

This manual contains, first, instructions for use of the new features of program AMP2 not found in program AMP. Section 2 together with the Antenna Modeling Program Users Manual (reference 1) should provide all information necessary for use of the program. The remainder of the manual covers the formulation of the surface modeling method and details of the program coding which differ from the basic program AMP. These sections supplement the Antenna Modeling Program Engineering Manual (reference 2) and Systems Manual (reference 3) respectively. Included also is a complete list of the AMP2 code in section 7.

2.0 PROGRAM OPERATION

The basic information needed to use program AMP2 is contained in the AMP Users Manual (reference 1). This section contains supplementary instructions and information for using AMP2 to model surfaces. When used to model structures with wires only, the operation of program AMP2 is identical to that of program AMP. The one exception is that AMP2 uses a time saving approximation in filling the interaction matrix for interaction distances greater than one wavelength. For results identical to those of AMP in all digits printed, this approximation range should be increased to greater than the maximum structure dimension in wavelengths by use of a KH card.

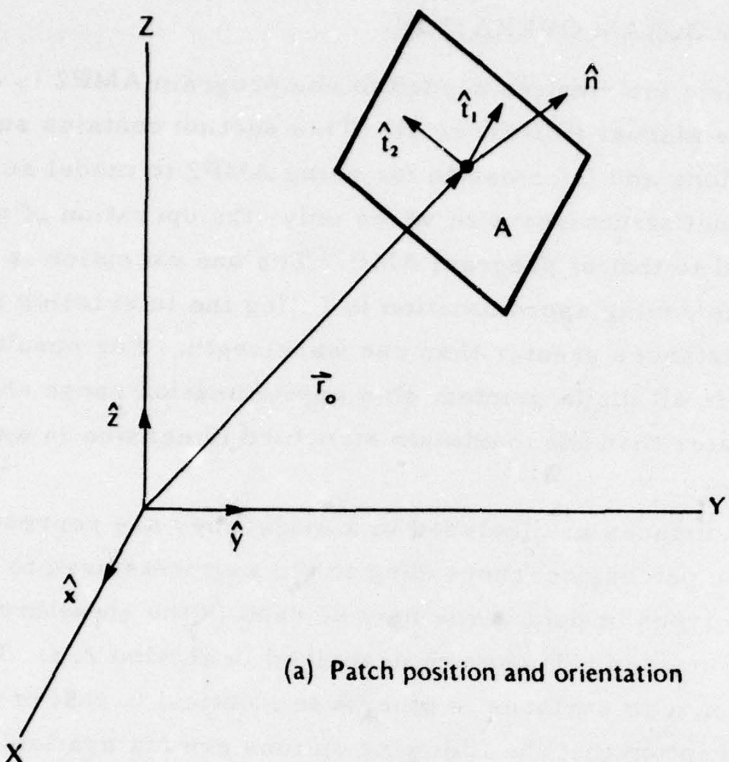
When surfaces are included in a model they are represented by small flat surface patches corresponding to the segments used to model wires. Two new types of data cards may be used in the structure geometry input to specify surfaces. These are described in section 2.3. The program operation with surfaces is otherwise identical to that of program AMP with the exception that the following options are not available when surfaces are modeled:

near field calculation
imperfect ground (perfect ground is allowed)
ground wave in the radiated field.

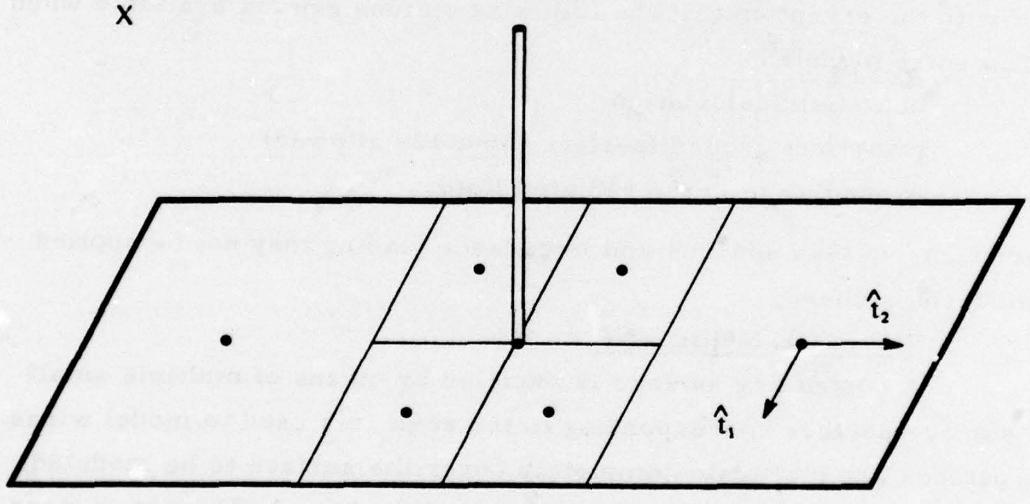
In addition, voltage sources and impedance loading may not be applied on surface patches.

2.1 SURFACE MODELING

A conducting surface is modeled by means of multiple small flat surface patches corresponding to the segments used to model wires. The patches are chosen to completely cover the surface to be modeled, conforming as closely as possible to curved surfaces. The parameters defining a surface patch are the Cartesian coordinates of the patch center, the components of the outward directed unit normal vector and the patch area. These are illustrated in figure 1(a) where $\overline{r_0} = x_0 \hat{x} + y_0 \hat{y} + z_0 \hat{z}$ is the position of the segment center, $\hat{n} = n_x \hat{x} + n_y \hat{y} + n_z \hat{z}$ is the unit normal vector and A is the patch area. The shape of



(a) Patch position and orientation



(b) Connection of a wire to a surface patch

FIGURE 1
SURFACE PATCH GEOMETRY

the patch (square, rectangular, etc.) is not specified since there is no integration over the patch unless a wire is connected to the center. The program computes the surface current on each patch along the orthogonal unit vectors $\hat{t}_1$ and $\hat{t}_2$ which are tangent to the surface. These are chosen by the program according to the following rules:

1. For a horizontal patch

$$\hat{t}_1 = \hat{x}$$

$$\hat{t}_2 = \hat{y}$$

2. For a non-horizontal patch

$$\hat{t}_1 = (\hat{z} \times \hat{n}) / |\hat{z} \times \hat{n}|$$

$$\hat{t}_2 = \hat{n} \times \hat{t}_1$$

When a structure having plane symmetry is formed by reflection in a coordinate plane using a GX input card (see reference 1) the vectors $\hat{t}_1$ $\hat{t}_2$ and $\hat{n}$ are also reflected so that the new patches will have $\hat{t}_2 = -\hat{n} \times \hat{t}_1$.

When a wire is connected to a surface the wire must end at the center of a patch with identical coordinates used for the wire end and the patch center. The program then divides the patch into four equal patches about the wire end as shown in Figure 1(b) where a wire has been connected to the second of three previously identical patches. The connection patch is divided along lines defined by the vectors $\hat{t}_1$ and $\hat{t}_2$ for that patch with a square patch assumed. The four new patches are ordinary patches like those input by the user, except when the interactions between these patches and the lowest segment on the connected wire are computed. In this case an interpolation function is applied to the four patches to represent the current from the wire onto the surface, and the function is numerically integrated over the patches. Thus, the shape of the patch is significant in this case. The user should try to choose patches so that those with wires connected are approximately square with sides parallel to $\hat{t}_1$ and $\hat{t}_2$. The connected wire is not required to be normal to the patch, but cannot lie in the plane of the patch.

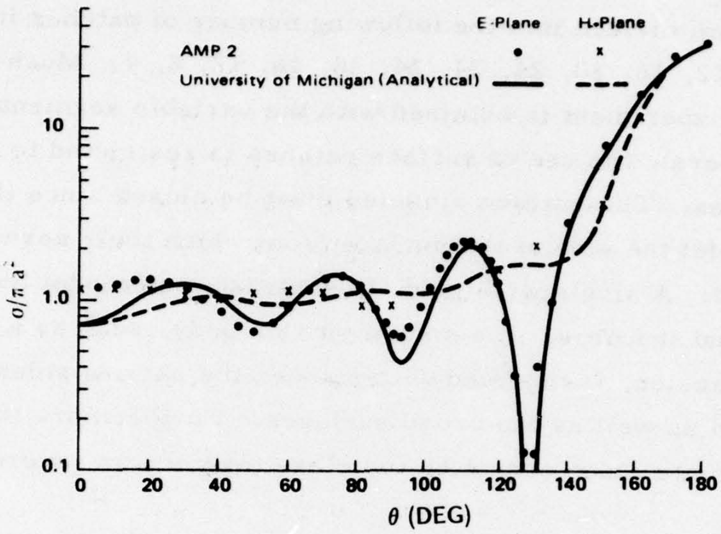
As with wire modeling, patch size measured in wavelengths is very important for accuracy of the results. A minimum of about 25 patches should be used per square wavelength of surface area, with the maximum size for an individual patch about 0.04 square wavelengths. Large patches may be used on large smooth surfaces while smaller patches are needed in areas of small radius of curvature, both for geometrical modeling accuracy and for accuracy of the integral operation solution. For the specific case of an edge, a precise local representation cannot be included; however, smaller patches in the vicinity of the edge can lead to more accurate results since the current magnitude may vary rapidly in this region. Since connection of a wire to a patch causes the patch to be divided into four smaller patches a larger patch may be input in anticipation of the subdivision.

While patch shape is not input to the program, very long narrow patches should be avoided when subdividing the surface. This is illustrated by the two methods of modeling a sphere shown below.

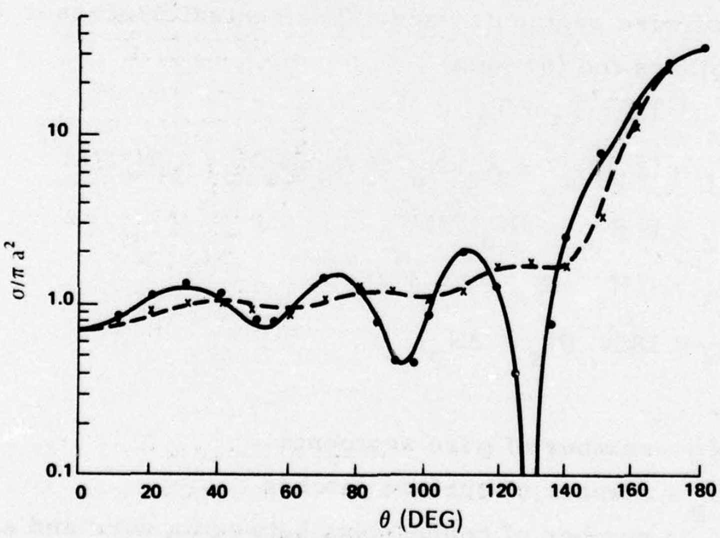
Uniform segmentation

Variable segmentation

The first uses uniform divisions in azimuth and equal cuts along the vertical axis. This results in all patches having equal areas but with long narrow patches near the poles. In the second method the number of divisions in azimuth is increased toward the equator so that the patch length and width are kept more nearly equal. The areas are again kept approximately equal. The results of the two segmentations are shown in Figure 2 for scattering by a sphere of ka (2πx radius/wavelength) equal



(a) Uniform Segmentation



(b) Variable ϕ Segmentation

FIGURE 2
BISTATIC RCS OF A SPHERE WITH ka = 5.3

MBA

to 5.3. The uniform segmentation used 14 increments in azimuth and 14 equal bands along the vertical axis. The variable segmentation used 13 equal increments in arc length along the vertical axis with each band from top to bottom divided into the following number of patches in azimuth: 4, 8, 12, 16, 20, 24, 24, 24, 20, 16, 12, 8, 4. Much better agreement with experiment is obtained with the variable segmentation.

In general, the use of surface patches is restricted to modeling voluminous bodies. The surface modeled must be closed since the patches only model the side of the surface from which their normals are directed outward. A single wire grid, in contrast, can model both sides of the surface that it forms. If a somewhat thin body, such as a box with one narrow dimension, is modeled with patches the narrow sides (edges) must be modeled as well as the broad surfaces. Furthermore the parallel surfaces on opposite sides cannot be too close together or severe numerical error will occur.

2.2 EXECUTION TIME

The program execution time depends on the number of patches and the number of wire segments used. The central processor time approximately follows the formula

$$T = T_1 + T_2 + T_3 + T_4$$

$$T_1 = (A_1 k N_s^2 + A_2 k N_p^2 + A_3 k N_s N_p + A_4 N_c)/M$$

$$T_2 = B\,(N_s + 2N_p)^3/M^2$$

$$T_3 = C N_e \cdot (N_s + 2N_p)^2/M$$

$$T_4 = D k N_f\,(N_s + 2N_p)$$

where

N_s = number of wire segments

N_p = number of surface patches

N_c = number of connections between a wire and a surface

N_e = number of different excitations

N_f = number of far field calculation points

M = number of degrees of symmetry

k = 1 if structure is in free space

2 if structure is over ground

T_1 is the time to fill the interaction matrix; T_2 is the time to factor the matrix; T_3 is the time to solve for the currents for all excitations and T_4 is the time to calculate far fields.

The proportionality factors depend on the computer system on which the program is run. The factors in seconds for a CDC 6600 computer with the program compiled under the Run compiler and the matrix fitting in core are roughly

$A_1 = 2. \ (10^{-3})$
$A_2 = 3. \ (10^{-4})$
$A_3 = 3. \ (10^{-3})$
$A_4 = 1. \ (10^{-1})$
$B \ = 5. \ (10^{-6})$
$C \ = 2. \ (10^{-5})$
$D \ = 3. \ (10^{-4})$

Unless a large number of excitations or far fields are requested, T_1 and T_2 will account for nearly all of the running time. If the matrix does not fit in core T_1 and T_2 will be larger than indicated above.

If the matrix fill approximation is used for interaction distances greater than R_0 (RKH = R_0/λ) then A_1 is multiplied by $(1. - 0.7\ R_w)$ and A_3 is multiplied by $(1. - 0.5\ R_p)$ where R_w is the fraction of all segment pairs for which the separation is greater than R_0 and R_p is the fraction of all segment-patch pairs for which the separation is greater than R_0.

2.3 NEW INPUT CARDS

All input cards described in the AMP User's Manual may be used in program AMP2. In addition two new cards have been added for input of surface patches, one card for selecting the distance at which the approximate interaction formula is applied, and one for setting the time interval between precautionary dumps of intermediate results to permit restarting the computation. When surface patches are input the cards GM, GX, GR and GS for moving duplicating or scaling a structure act on patches as well as on wire segments as described in reference 1. All patches input before the GM, GX, GR or GS card is encountered will be affected. Since patches do not have the tag numbers that segments have the GM card must act on all patches input before its occurrence.

The number of patches and segments that may be used in a model is limited by program dimensions. If the number of segments is N_s and the number of patches N_p then the standard limits in the program are

$$N_s + N_p \leq 1000 \text{ if } N_p \leq 500$$

$$N_s + 3\,N_p \leq 2000 \text{ if } N_p > 500$$

The new input cards are described on the following pages.

SURFACE PATCH (SP)

PURPOSE: to input the parameters of a single surface patch

CARD:

2	5	10	20	30	40	50	60	70	80
SP	Blank	Blank	F1	F2	F3	F4	F5	F6	Blank
			XC	YC	ZC	AL	BT	AR	

The numbers along the top refer to the last column in each field.

PARAMETERS:

INTEGERS - None

DECIMAL NUMBERS

XC (F1) - } x, y and z coordinates
YC (F2) - } of the center of the
ZC (F3) - } patch

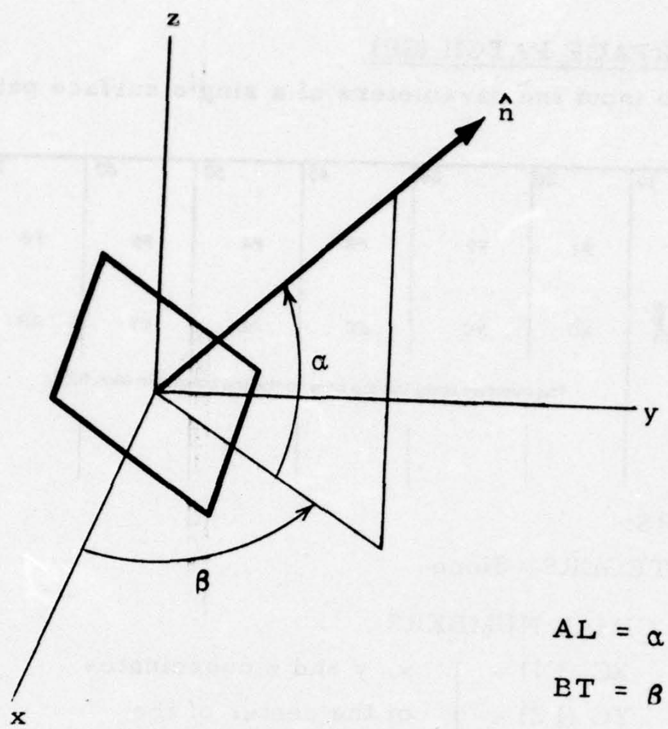
AL (F4) - elevation angle above the x - y plane of the outward normal to the patch (degrees)

BT (F5) - angle from the x axis to the projection of the outward normal of the patch onto the x - y plane (degrees)

AR (F6) - area of the patch (units are the square of the units used for XC, YC, and ZC)

NOTES:

- The use of AL and BT in defining the normal to a patch is illustrated in the figure below. For a horizontal patch AL = 90. and BT = 0.



NOTES:

- SP cards, if used, must occur in the section of geometry data cards -- after the comment cards and before the GE card. They may be intermixed with cards specifying wires and any other geometry data cards.
- At the end of structure geometry input the patch coordinates, like the wire coordinates, must be in meters. If other units are used for input they may be scaled by use of a GS card which scales both patch dimensions and wire dimensions.

MULTIPLE SURFACE PATCHES (SS)

PURPOSE: to cover a flat surface with multiple patches by reproducing the previous patch input with shifts in the X and Y directions.

CARD:

2	5	10	20	30	40	50	60	70	80
SS	I1	I2	F1	F2	Blank	Blank	Blank	Blank	Blank
	NX	NY	DX	DY					

The numbers along the top refer to the last column in each field.

PARAMETERS:

INTEGERS

NX (I1) - the previous patch input is reproduced NX times with increments in the X direction

NY (I2) - the previous patch input is reproduced NY times with increments in the Y direction

DECIMAL NUMBERS

DX (F1) - X coordinate increment for reproduced patches

DY (F2) - Y coordinate increment for reproduced patches

NOTES:

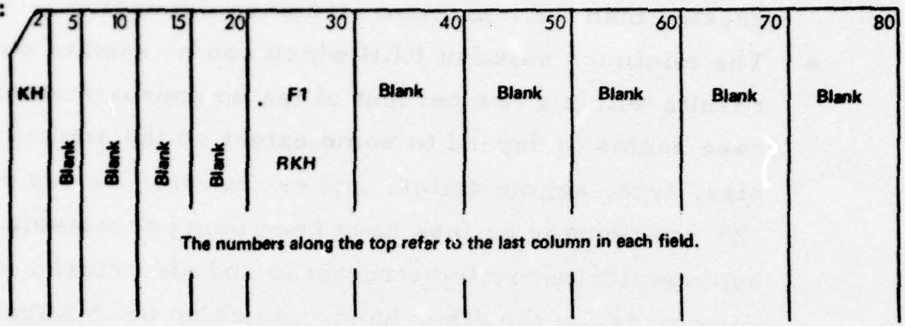
- The surface generated by a SS card is NX + 1 patches wide in the X direction and NY + 1 patches wide in the Y direction The patch is first reproduced NX times in the X direction with increment DX. The Y coordinate is then incremented and NX + 1 patches are generated adjacent to the first row in the X direction. This process is repeated NY times. The patch reproduced may have any orientation (which is maintained in the new patches) so the SS card may be used to generate both the top and sides of a box. The SS card will not shift patches in the Z direction, however.

- The increments DX and DY must be consistent with the area of the patch reproduced so that the sum of the areas of all patches is equal to the area of the total surface covered.
- The GM card may also be used to reproduce patches with an arbitrary direction of shift but it operates on all patches and wires input before the GM card.

INTERACTION APPROXIMATION RANGE (KH)

PURPOSE: to set the minimum separation distance for use of a time saving approximation in filling the interaction matrix.

CARD:



PARAMETERS:

INTEGERS - none

DECIMAL NUMBERS

RKH (F1) - The approximation is used for interactions over distances greater than RKH wavelengths.

NOTES:

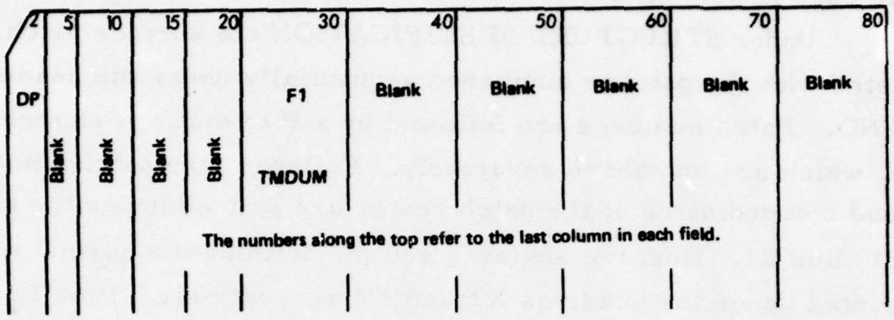
- *If two segments* or a segment and a patch are separated by more than RKH wavelengths the interaction field is computed from an impulse approximation to the segment current. The field of a current element located at the segment center is used. For separations less than RKH a current interpolation function is integrated over the segment length as in the basic AMP program. No approximation is used for the field due to the surface current on a patch since the time for the standard calculation is very short.
- The KH card can be placed anywhere in the data cards following the geometry cards (with FR, EX, LD, etc.) and affects all calculations requested following its occurrence. The value of RKH may be changed within a data set by use of a new KH card.

- If no KH card is used RKH has a default value of one wavelength. Hence to exactly duplicate a run with the basic program AMP a KH card should be used with RKH greater than the maximum structure dimension.
- The minimum value of RKH which can be used to obtain results within a few percent of the no approximation case seems to depend to some extent on the structure size, type, segmentation, and excitation. Values of .25 wavelengths or less have been found acceptable for symmetrically excited structures and electrically small wire grids; on the other hand, values up to .5 wavelengths have been required for very asymmetrically fed structures. No exact guidelines have been developed for RKH; therefore, it is best to experiment on any given problem type if a minimum value is desired. RKH should never be less than the length of the longest segment, however.

FILE DUMP TIME INTERVAL (DP)

PURPOSE: to set the time interval between automatic dumps of file storage to permit restarting the program after a machine failure.

CARD:



PARAMETERS:

INTEGERS - None

DECIMAL NUMBERS

TMDUM (F1) - Time interval in seconds between dumps of file storage

NOTES:

- Use of a DP card will produce a dump of the program files every TMDUM seconds during the filling and factoring of the interaction matrix. These are the most time consuming operations in the program. After the matrix has been factored no additional dumps will occur. If the structure being run is small enough to run without file storage no dumps will occur.
- File 17 to which the files are dumped must be requested as a magnetic tape or other permanent storage device by a control card.
- Instructions for restarting a run from the file dump are given in Appendix B of reference 1.

2.4 PROGRAM OUTPUT

The program output is basically the same as that for program AMP, described in reference 1, with some additional data printed for surface patches. The new data pertaining to patches is illustrated in the sample case in section 2.5.

Under STRUCTURE SPECIFICATION the surface patch cards are listed with the patches numbered sequentially under the heading WIRE NO. Patch numbers are followed by a P to distinguish them from wires, which are numbered separately. Following the patch number, the x, y and z coordinates of the patch center are printed under the headings X1, Y1, and Z1. Next the angles α and β, defining the normal to the patch, are printed under the headings X2 and Y2 respectively. Finally the patch area is printed under the heading RADIUS. The other columns used for wire data are blank for patches.

Following SEGMENTATION DATA, a block labeled SURFACE PATCH DATA is printed with the x, y and z coordinates of the patch center; the x, y and z components of the unit normal vector and the patch area. Also printed are the x, y and z components of the unit tangent vector $\hat{t}_1$ under the headings X1, Y1 and Z1, and $\hat{t}_2$ under the headings X2, Y2, and Z2.

Following the printing of frequency the value of RKH is printed in the format

APPROXIMATE INTEGRATION EMPLOYED FOR SEGMENTS MORE THAN "RKH" WAVELENGTHS APART.

Finally, following the segment current data, is a block labeled SURFACE PATCH CURRENTS. This lists, for each patch, the coordinates of the patch center in wavelengths, the patch area in square wavelengths and the surface current density in amps per meter. The surface current is given both as surface components along the vectors $\hat{t}_1$ and $\hat{t}_2$ in magnitude and phase, and as x, y and z components in real and imaginary parts. The remainder of the output is the same as for the standard program AMP.

2.5 SAMPLE CASE

As an example of the use of the program for surfaces and wires the input cards and resulting output are shown on the following pages for a cylinder with a wire antenna and another parasitic wire element attached.

The cylinder is generated by first specifying three patches in a column centered on the x axis as shown in figure 3(a). A GM card is then used to produce a second column of patches rotated about the z axis by 30 degrees. A patch is then added to the top and another to the bottom forming parts of the end surfaces. The model at this point is shown in figure 3(b). Next a GR card is used to rotate this section of patches about the z axis to form a total of six similar sections, including the original. A patch is then added to the center of the top and another to the bottom to form the complete cylinder shown in figure 3(c). Finally two GW cards are used to add wires connecting to the top and side of the cylinder. The patches to which the wires are connected are divided into four smaller patches as shown in figure 3(d). Although patch shape is not input to the program, square patches are assumed at the base of a connected wire when integrating over the surface current. Hence a more accurate representation of the model would be as shown in figure 4 where the patches to which wires connect are square with equal areas maintained for all patches (before subdivision). The remaining data cards scale the structure by a factor of 0.01, specify a frequency of 465.84 MHz, specify a unit voltage source at the base of the wire on the top of the cylinder and request computation of a radiation pattern.

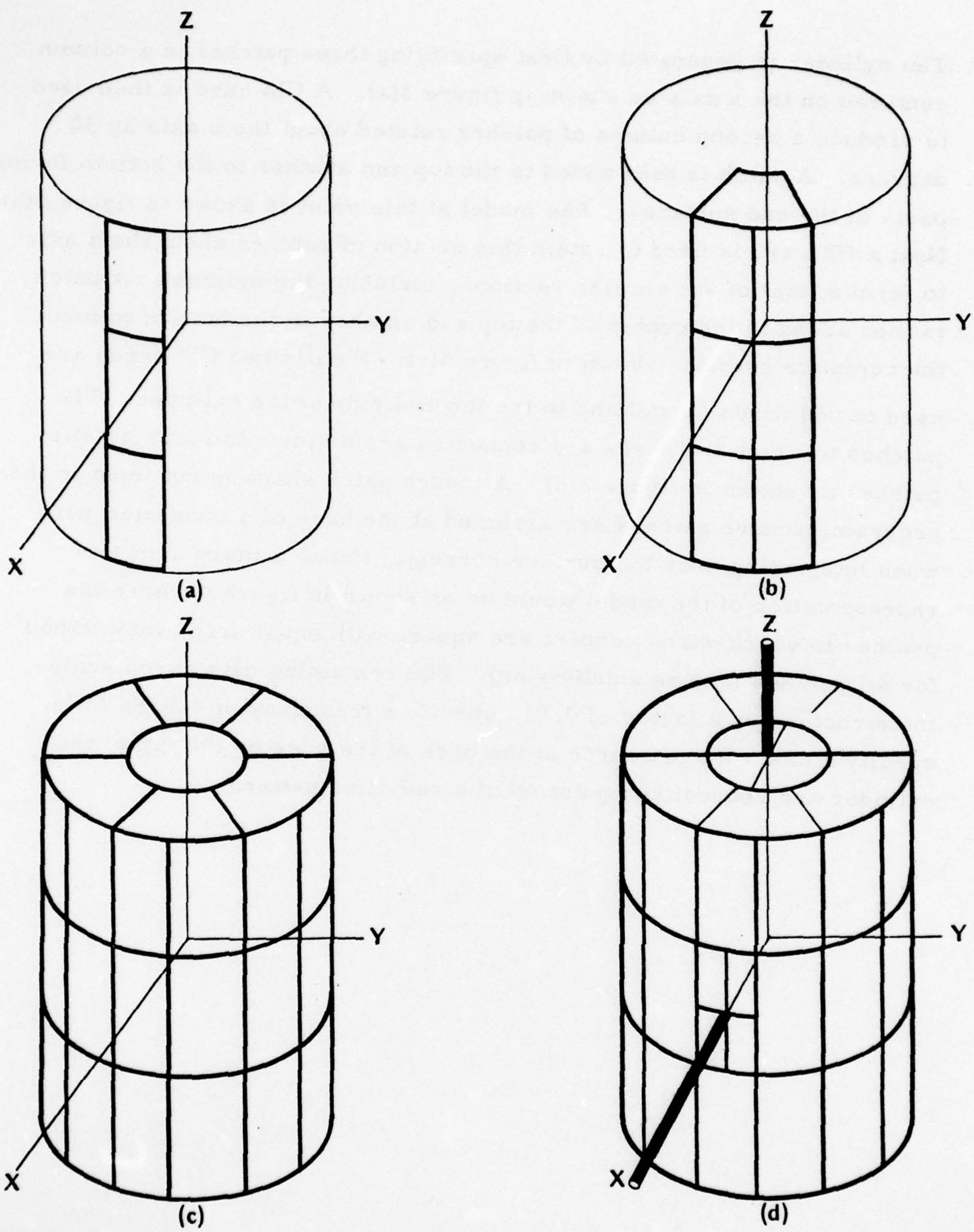


FIGURE 3
DEVELOPMENT OF SURFACE MODEL FOR CYLINDER WITH ATTACHED WIRES

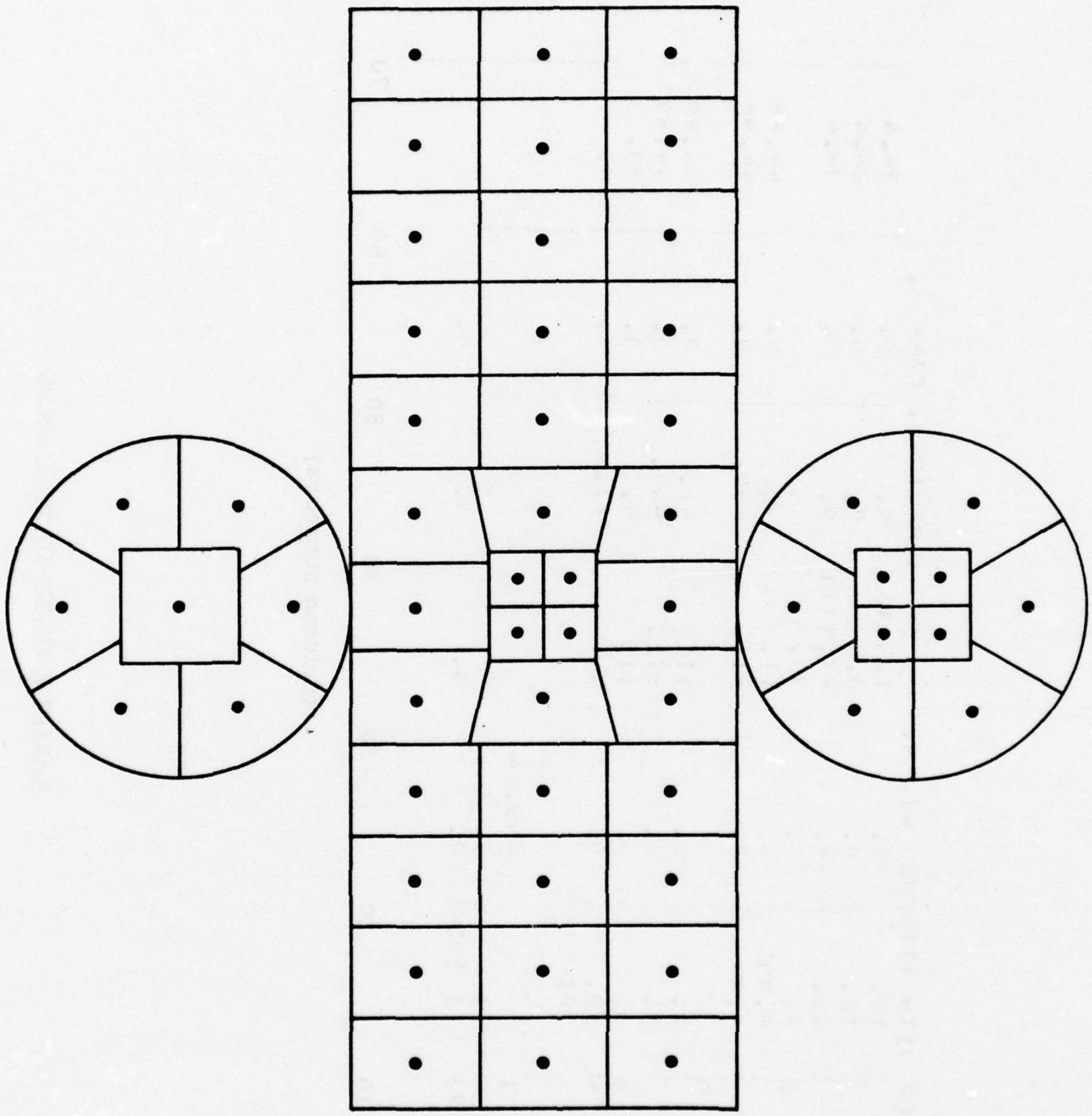


FIGURE 4
SEGMENTATION OF CYLINDER FOR WIRES CONNECTED TO END AND SIDE

MBA

```
CECYLINDER WITH ATTACHED WIRES.             ALBERTSEN. CASE 8.
SP          10.       0.        7.3333    0.        0.        38.4
SP          10.       0.        0.        0.        0.        38.4
SP          10.       0.        -7.3333   0.        0.        38.4
GM        1 0.        0.        30.
SP          6.89      0.        11.       90.       0.        44.88
SP          6.89      0.        -11.      -90.      0.        44.88
GR        6
SP          0.        0.        11.       90.       0.        44.89
SP          0.        0.        -11.      -90.      0.        44.89
GW        4 0.        0.        11.       0.        0.        23.       .1
GW        5 10.       0.        0.        27.6      0.        0.        .2
GS          .01
GE
FR        1           465.94
EX            1       1.
RP       73   1 1000  0.        0.        5.        0.
EN
2        10        20        30        40        50        60        70        80
```

(column numbers)

SAMPLE CASE DATA CARDS

****

ANTENNA MODELING PROGRAM

****

- - - - COMMENTS - - - -

CYLINDER WITH ATTACHED WIRES. ALBERTSEN, CASE A.

- - - STRUCTURE SPECIFICATION - - -

COORDINATES MUST BE INPUT IN
METERS OR BE SCALED TO METERS
BEFORE STRUCTURE INPUT IS ENDED

WIRE NO.	X1	Y1	Z1	X2	Y2	Z2	RADIUS	NO. OF SEG.	FIRST SEG.	LAST SEG.	TAG NO.
1P	10.00000	0.00000	7.33330	0.00000	0.00000		38.40000				
2P	10.00000	0.00000	0.00000	0.00000	0.00000		38.40000				
3P	10.00000	0.00000	-7.33330	0.00000	0.00000		38.40000				

THE STRUCTURE HAS BEEN MOVED. MOVE DATA CARD IS -
-0 1 0.00000 0.00000 30.00000 -0.00000 -0.00000 -0.00000 -0.00000

WIRE NO.	X1	Y1	Z1	X2	Y2	Z2	RADIUS	NO. OF SEG.	FIRST SEG.	LAST SEG.	TAG NO.
7P	6.89000	0.00000	11.00000	90.00000	0.00000		44.88000				
8P	6.89000	0.00000	-11.00000	-90.00000	0.00000		44.88000				

STRUCTURE ROTATED ABOUT Z-AXIS 8 TIMES. LABLES INCREMENTED BY -0

WIRE NO.	X1	Y1	Z1	X2	Y2	Z2	RADIUS	NO. OF SEG.	FIRST SEG.	LAST SEG.	TAG NO.
49P	0.00000	0.00000	11.00000	90.00000	0.00000		44.89000				
50P	0.00000	0.00000	-11.00000	-90.00000	0.00000		44.89000				
1	0.00000	0.00000	11.00000	0.00000	0.00000	23.00000	.10000	4	1	4	-0
2	10.00000	0.00000	0.00000	27.60000	0.00000	0.00000	.20000	5	5	9	-0

STRUCTURE SCALED BY FACTOR .01000

TOTAL SEGMENTS USED= 9 NO. SEG. IN A SYMMETRIC CELL= 9 SYMMETRY FLAG= 0
TOTAL PATCHES USED= 56 NO. PATCHES IN A SYMMETRIC CELL= 56

- - - - SEGMENTATION DATA - - - -

COORDINATES IN METERS

I+ AND I- INDICATE THE SEGMENTS BEFORE AND AFTER I

SEG. NO.	COORDINATES OF SEG. CENTER X	Y	Z	SEG. LENGTH	ORIENTATION ANGLES ALPHA	BETA	WIRE RADIUS	CONNECTION DATA I-	I	I+	TAG NO.
1	0.00000	0.00000	.12500	.03000	90.00000	0.00000	.00100	10052	1	2	-0
2	0.00000	0.00000	.15500	.03000	90.00000	0.00000	.00100	1	2	3	-0
3	0.00000	0.00000	.18500	.03000	90.00000	0.00000	.00100	2	3	4	-0
4	0.00000	0.00000	.21500	.03000	90.00000	0.00000	.00100	3	4	0	-0
5	.11760	0.00000	0.00000	.03520	0.00000	0.00000	.00200	10002	5	6	-0
6	.15280	0.00000	0.00000	.03520	0.00000	0.00000	.00200	5	6	7	-0
7	.18800	0.00000	0.00000	.03520	0.00000	0.00000	.00200	6	7	8	-0
8	.22320	0.00000	0.00000	.03520	0.00000	0.00000	.00200	7	8	9	-0
9	.25840	0.00000	0.00000	.03520	0.00000	0.00000	.00200	8	9	0	-0

- - - SURFACE PATCH DATA - - -

COORDINATES IN METERS

PATCH NO.	COORDINATES OF PATCH CENTER X	Y	Z	UNIT NORMAL VECTOR X	Y	Z	PATCH AREA	COMPONENTS OF UNIT TANGENT VECTORS X1	Y1	Z1	X2	Y2	Z2
1	.10000	0.00000	.07333	1.0000	0.0000	0.0000	.00384	-0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
2	.10000	.01549	.01549	1.0000	0.0000	0.0000	.00096	-0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
3	.10000	-.01549	.01549	1.0000	0.0000	0.0000	.00096	-0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
4	.10000	-.01549	-.01549	1.0000	0.0000	0.0000	.00096	-0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
5	.10000	.01549	-.01549	1.0000	0.0000	0.0000	.00096	-0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
6	.10000	0.00000	-.07333	1.0000	0.0000	0.0000	.00384	-0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
7	.08660	.05000	.07333	.8660	.5000	0.0000	.00384	-.5000	.8660	0.0000	0.0000	0.0000	1.0000
8	.08660	.05000	0.00000	.8660	.5000	0.0000	.00384	-.5000	.8660	0.0000	0.0000	0.0000	1.0000
9	.08660	.05000	-.07333	.8660	.5000	0.0000	.00384	-.5000	.8660	0.0000	0.0000	0.0000	1.0000
10	.06890	0.00000	.11000	0.0000	0.0000	1.0000	.00449	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000
11	.06890	0.00000	-.11000	0.0000	0.0000	-1.0000	.00449	1.0000	0.0000	0.0000	0.0000	-1.0000	0.0000
12	.05000	.08660	.07333	.5000	.8660	0.0000	.00384	-.8660	.5000	0.0000	0.0000	0.0000	1.0000
13	.05000	.08660	0.00000	.5000	.8660	0.0000	.00384	-.8660	.5000	0.0000	0.0000	0.0000	1.0000
14	.05000	.08660	-.07333	.5000	.8660	0.0000	.00384	-.8660	.5000	0.0000	0.0000	0.0000	1.0000
15	-.00000	.10000	.07333	-.0000	1.0000	0.0000	.00384	-1.0000	-.0000	0.0000	0.0000	0.0000	1.0000
16	-.00000	.10000	0.00000	-.0000	1.0000	0.0000	.00384	-1.0000	-.0000	0.0000	0.0000	0.0000	1.0000
17	-.00000	.10000	-.07333	-.0000	1.0000	0.0000	.00384	-1.0000	-.0000	0.0000	0.0000	0.0000	1.0000
18	.03445	.05967	.11000	0.0000	0.0000	1.0000	.00449	.5000	.8660	0.0000	-.8660	.5000	0.0000
19	.03445	.05967	-.11000	0.0000	0.0000	-1.0000	.00449	.5000	.8660	0.0000	.8660	-.5000	0.0000
20	-.05000	.08660	.07333	-.5000	.8660	0.0000	.00384	-.8660	-.5000	0.0000	0.0000	0.0000	1.0000
21	-.05000	.08660	0.00000	-.5000	.8660	0.0000	.00384	-.8660	-.5000	0.0000	0.0000	0.0000	1.0000
22	-.05000	.08660	-.07333	-.5000	.8660	0.0000	.00384	-.8660	-.5000	0.0000	0.0000	0.0000	1.0000
23	-.08660	.05000	.07333	-.8660	.5000	0.0000	.00384	-.5000	-.8660	0.0000	0.0000	0.0000	1.0000
24	-.08660	.05000	0.00000	-.8660	.5000	0.0000	.00384	-.5000	-.8660	0.0000	0.0000	0.0000	1.0000
25	-.08660	.05000	-.07333	-.8660	.5000	0.0000	.00384	-.5000	-.8660	0.0000	0.0000	0.0000	1.0000
26	-.03445	.05967	.11000	0.0000	0.0000	1.0000	.00449	-.5000	.8660	0.0000	-.8660	-.5000	0.0000
27	-.03445	.05967	-.11000	0.0000	0.0000	-1.0000	.00449	-.5000	.8660	0.0000	.8660	.5000	0.0000
28	-.10000	-.00000	.07333	-1.0000	-.0000	0.0000	.00384	.0000	-1.0000	0.0000	0.0000	0.0000	1.0000
29	-.10000	-.00000	0.00000	-1.0000	-.0000	0.0000	.00384	.0000	-1.0000	0.0000	0.0000	0.0000	1.0000
30	-.10000	-.00000	-.07333	-1.0000	-.0000	0.0000	.00384	.0000	-1.0000	0.0000	0.0000	0.0000	1.0000
31	-.08660	-.05000	.07333	-.8660	-.5000	0.0000	.00384	.5000	-.8660	0.0000	0.0000	0.0000	1.0000
32	-.08660	-.05000	0.00000	-.8660	-.5000	0.0000	.00384	.5000	-.8660	0.0000	0.0000	0.0000	1.0000
33	-.08660	-.05000	-.07333	-.8660	-.5000	0.0000	.00384	.5000	-.8660	0.0000	0.0000	0.0000	1.0000
34	-.06890	-.00000	.11000	0.0000	0.0000	1.0000	.00449	-1.0000	-.0000	0.0000	.0000	-1.0000	0.0000
35	-.06890	-.00000	-.11000	0.0000	0.0000	-1.0000	.00449	-1.0000	-.0000	0.0000	-.0000	1.0000	0.0000
36	-.05000	-.08660	.07333	-.5000	-.8660	0.0000	.00384	.8660	-.5000	0.0000	0.0000	0.0000	1.0000
37	-.05000	-.08660	0.00000	-.5000	-.8660	0.0000	.00384	.8660	-.5000	0.0000	0.0000	0.0000	1.0000
38	-.05000	-.08660	-.07333	-.5000	-.8660	0.0000	.00384	.8660	-.5000	0.0000	0.0000	0.0000	1.0000
39	.00000	-.10000	.07333	.0000	-1.0000	0.0000	.00384	1.0000	.0000	0.0000	0.0000	0.0000	1.0000
40	.00000	-.10000	0.00000	.0000	-1.0000	0.0000	.00384	1.0000	.0000	0.0000	0.0000	0.0000	1.0000
41	.00000	-.10000	-.07333	.0000	-1.0000	0.0000	.00384	1.0000	.0000	0.0000	0.0000	0.0000	1.0000
42	-.03445	-.05967	.11000	0.0000	0.0000	1.0000	.00449	-.5000	-.8660	0.0000	.8660	-.5000	0.0000
43	-.03445	-.05967	-.11000	0.0000	0.0000	-1.0000	.00449	-.5000	-.8660	0.0000	-.8660	.5000	0.0000
44	.05000	-.08660	.07333	.5000	-.8660	0.0000	.00384	.8660	.5000	0.0000	0.0000	0.0000	1.0000
45	.05000	-.08660	0.00000	.5000	-.8660	0.0000	.00384	.8660	.5000	0.0000	0.0000	0.0000	1.0000
46	.05000	-.08660	-.07333	.5000	-.8660	0.0000	.00384	.8660	.5000	0.0000	0.0000	0.0000	1.0000
47	.08660	-.05000	.07333	.8660	-.5000	0.0000	.00384	.5000	.8660	0.0000	0.0000	0.0000	1.0000
48	.08660	-.05000	0.00000	.8660	-.5000	0.0000	.00384	.5000	.8660	0.0000	0.0000	0.0000	1.0000
49	.08660	-.05000	-.07333	.8660	-.5000	0.0000	.00384	.5000	.8660	0.0000	0.0000	0.0000	1.0000
50	.03445	-.05967	.11000	0.0000	0.0000	1.0000	.00449	.5000	-.8660	0.0000	.8660	.5000	0.0000
51	.03445	-.05967	-.11000	0.0000	0.0000	-1.0000	.00449	.5000	-.8660	0.0000	-.8660	-.5000	0.0000

52	.01675	.01675	.11000	0.0000	0.0000	1.0000	.00112	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000
53	-.01675	.01675	.11000	0.0000	0.0000	1.0000	.00112	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000
54	-.01675	-.01675	.11000	0.0000	0.0000	1.0000	.00112	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000
55	.01675	-.01675	.11000	0.0000	0.0000	1.0000	.00112	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000
56	0.00000	0.00000	-.11000	0.0000	0.0000	-1.0000	.00449	1.0000	0.0000	0.0000	0.0000	-1.0000	0.0000

```
***** DATA CARD NO.  1  FR  -0    1   -0   -0  4.65840E+02 -0.            -0.          -0.          -0.          -0.
***** DATA CARD NO.  2  EX  -0   -0    1   -0  1.00000E+00 -0.            -0.          -0.          -0.          -0.
***** DATA CARD NO.  3  RP  -0   73    1 1000  0.            0.     5.00000E+00  0.          -0.          -0.
```

- - - - - - FREQUENCY - - - - - -

FREQUENCY= 4.6584E+02 MHZ
WAVELENGTH= 6.4400E-01 METERS

APPROXIMATE INTEGRATION EMPLOYED FOR SEGMENTS MORE THAN 1.000 WAVELENGTHS APART

- - - STRUCTURE IMPEDANCE LOADING - - -

THIS STRUCTURE IS NOT LOADED

- - - ANTENNA ENVIRONMENT - - -

FREE SPACE

- - - MATRIX TIMING - - -

FILL= 3.161 SEC., FACTOR= 11.573 SEC.

- - - ANTENNA INPUT PARAMETERS - - -

TAG NO.	SEG. NO.	VOLTAGE (VOLTS) REAL	VOLTAGE (VOLTS) IMAG.	CURRENT (AMPS) REAL	CURRENT (AMPS) IMAG.	IMPEDANCE (OHMS) REAL	IMPEDANCE (OHMS) IMAG.	ADMITTANCE (MHOS) REAL	ADMITTANCE (MHOS) IMAG.	POWER (WATTS)
-0	1	1.00000E+00	-0.	1.11302E-03	7.74029E-03	1.80184E+01	-1.25952E+02	1.11302E-03	7.74029E-03	5.56512E-04

- - - CURRENTS AND LOCATION - - -

DISTANCES IN WAVELENGTHS

SEG. NO.	TAG NO.	COORD. OF SEG. CENTER X	Y	Z	SEG. LENGTH	CURRENT (AMPS) REAL	IMAG.	MAG.	PHASE
1	-0	0.0000	0.0000	.1941	.04658	1.1130E-03	7.7803E-03	7.8595E-03	81.859
2	-0	0.0000	0.0000	.2407	.04658	9.7807E-04	6.0893E-03	6.1674E-03	80.875
3	-0	0.0000	0.0000	.2873	.04658	7.0300E-04	4.0343E-03	4.0951E-03	80.115
4	-0	0.0000	0.0000	.3339	.04658	2.9340E-04	1.5855E-03	1.6124E-03	79.516
5	-0	.1926	0.0000	0.0000	.05466	-8.7268E-04	1.5334E-03	1.7644E-03	119.645
6	-0	.2373	0.0000	0.0000	.05466	-8.0623E-04	1.4135E-03	1.6272E-03	119.700
7	-0	.2419	0.0000	0.0000	.05466	-6.7010E-04	1.1750E-03	1.3526E-03	119.696
8	-0	.3466	0.0000	0.0000	.05466	-4.6441E-04	8.1639E-04	9.3924E-04	119.634
9	-0	.4012	0.0000	0.0000	.05466	-1.9136E-04	3.3792E-04	3.8834E-04	119.523

- - - - SURFACE PATCH CURRENTS - - - -

DISTANCE IN WAVELENGTHS
CURRENT IN AMPS/METER

PATCH NO.	PATCH CENTER X	Y	Z	PATCH AREA	SURFACE COMPONENTS: TANGENT VECTOR 1 MAG.	PHASE	TANGENT VECTOR 2 MAG.	PHASE	RECTANGULAR COMPONENTS: X REAL	X IMAG.	Y REAL	Y IMAG.	Z REAL	Z IMAG.
1	.155	0.000	.114	.00926	1.7142E-09	-166.95	7.0100E-03	-34.50	0.	0.	-1.57E-09	-3.87E-10	5.78E-03	-3.97E-03
2	.155	.024	.024	.00231	9.1709E-03	-61.78	1.3924E-02	-49.68	0.	0.	4.34E-03	-8.08E-03	9.01E-03	-1.06E-02
3	.155	-.024	.024	.00231	9.1709E-03	118.22	1.3924E-02	-49.68	0.	0.	-4.34E-03	8.08E-03	9.01E-03	-1.06E-02
4	.155	-.024	-.024	.00231	9.1694E-03	119.14	5.4495E-03	96.39	0.	0.	-4.33E-03	8.09E-03	-6.06E-04	5.41E-03
5	.155	.024	-.024	.00231	9.1694E-03	-61.86	5.4485E-03	96.39	0.	0.	4.33E-03	-8.09E-03	-6.06E-04	5.41E-03
6	.155	0.000	-.114	.00926	1.7135E-09	-166.70	1.1795E-03	37.39	0.	0.	-1.67E-09	-3.94E-10	9.37E-04	7.16E-04
7	.134	.078	.114	.00926	1.8492E-03	-89.95	5.2792E-03	-11.18	-8.29E-07	9.25E-04	1.44E-06	-1.60E-03	5.18E-03	-1.02E-03
8	.134	.078	0.000	.00926	3.4241E-03	-71.56	4.7337E-03	-32.10	-6.21E-04	1.86E-03	1.08E-03	-3.22E-03	4.01E-03	-2.52E-03
9	.134	.078	-.114	.00926	1.7234E-03	-93.79	2.3165E-03	-31.90	5.69E-05	8.60E-04	-9.86E-05	-1.49E-03	1.97E-03	-1.22E-03
10	.107	0.000	.171	.01082	1.2704E-02	-104.82	1.7943E-09	-167.70	-3.25E-03	-1.23E-02	-1.75E-09	-3.82E-10	0.	0.
11	.107	0.000	-.171	.01082	1.4185E-03	-8.41	1.7898E-09	12.41	1.40E-03	-2.07E-04	-1.75E-09	-3.85E-10	0.	0.
12	.078	.134	.114	.00926	1.5624E-03	-125.80	4.4093E-03	-8.25	7.92E-04	1.10E-03	-4.57E-04	-6.34E-04	4.36E-03	-6.33E-04
13	.078	.134	0.000	.00926	1.5414E-03	-110.78	4.4713E-03	-32.39	4.74E-04	1.25E-03	-2.73E-04	-7.21E-04	4.11E-03	-2.61E-03
14	.078	.134	-.114	.00926	1.6450E-03	-133.90	3.7035E-03	-48.41	1.00E-03	1.04E-03	-5.77E-04	-6.00E-04	2.46E-03	-2.77E-03
15	-.000	.155	.114	.00926	1.3423E-03	-159.88	4.3566E-03	12.21	1.26E-03	4.62E-04	1.74E-13	6.39E-14	4.84E-03	1.05E-03
16	-.000	.155	0.000	.00926	1.1687E-03	-158.23	4.9901E-03	-32.42	1.09E-03	4.33E-04	1.50E-13	6.00E-14	4.21E-03	-2.68E-03
17	-.000	.155	-.114	.00926	1.5473E-03	-167.81	4.2640E-03	-54.07	1.51E-03	3.27E-04	2.09E-13	4.52E-14	2.50E-03	-3.45E-03
18	.053	.093	.171	.01082	1.3817E-02	-104.86	1.4844E-03	-143.04	-7.45E-04	-5.40E-03	-3.66E-03	-1.20E-02	0.	0.
19	.053	.093	-.171	.01082	1.6588E-03	-49.57	1.9667E-03	31.54	1.69E-03	7.85E-05	2.64E-04	-1.50E-03	0.	0.
20	-.078	.134	.114	.00926	1.2522E-03	179.91	4.7391E-03	-2.85	1.08E-03	-1.78E-06	6.26E-04	-1.03E-06	4.73E-03	-2.35E-04
21	-.078	.134	0.000	.00926	1.0176E-03	175.06	5.1150E-03	-32.80	8.78E-04	-7.59E-05	5.07E-04	-4.38E-05	4.30E-03	-2.77E-03
22	-.078	.134	-.114	.00926	1.4961E-03	172.79	4.1035E-03	-55.66	1.28E-03	-1.61E-04	7.37E-04	-9.32E-05	2.31E-03	-3.39E-03
23	-.134	.078	.114	.00926	6.8901E-04	159.42	5.4126E-03	8.61	3.23E-04	-1.21E-04	5.54E-04	-2.10E-04	5.35E-03	8.10E-04
24	-.134	.078	0.000	.00926	6.0944E-04	162.20	5.2003E-03	-32.83	2.90E-04	-9.32E-05	5.03E-04	-1.61E-04	4.37E-03	-2.82E-03
25	-.134	.078	-.114	.00926	8.1991E-04	162.37	4.0730E-03	-57.63	3.90E-04	-1.24E-04	6.76E-04	-2.15E-04	2.18E-03	-3.44E-03
26	-.053	.093	.171	.01082	1.4481E-02	-108.64	1.1284E-03	-177.54	3.46E-03	6.91E-03	-3.34E-03	-1.19E-02	0.	0.
27	-.053	.093	-.171	.01082	1.6060E-03	-93.33	1.4794E-03	-1.42	1.33E-03	7.70E-04	6.59E-04	-1.41E-03	0.	0.
28	-.155	-.000	.114	.00926	1.6584E-09	34.80	5.0108E-03	-4.86	5.59E-19	3.88E-19	-1.36E-09	-9.46E-10	4.99E-03	-4.25E-04
29	-.155	-.000	0.000	.00926	5.5205E-08	102.42	5.2307E-03	-32.91	-4.87E-18	2.21E-17	1.19E-08	-5.39E-08	4.39E-03	-2.84E-03
30	-.155	-.000	-.114	.00926	1.6545E-09	34.66	3.9262E-03	-56.36	5.58E-19	3.86E-19	-1.36E-09	-9.41E-10	2.17E-03	-3.27E-03
31	-.134	-.078	.114	.00926	6.8901E-04	-20.58	5.4126E-03	8.61	3.23E-04	-1.21E-04	-5.59E-04	2.10E-04	5.35E-03	8.10E-04
32	-.134	-.078	0.000	.00926	6.0444E-04	-17.80	5.2003E-03	-32.83	2.90E-04	-9.32E-05	-5.03E-04	1.61E-04	4.37E-03	-2.82E-03
33	-.134	-.078	-.114	.00926	8.1992E-04	-17.63	4.0730E-03	-57.63	3.90E-04	-1.24E-04	-6.76E-04	2.15E-04	2.18E-03	-3.44E-03
34	-.107	-.000	.171	.01082	1.4438E-02	-111.06	2.6757E-09	61.11	5.19E-03	1.35E-02	-1.29E-09	-2.34E-09	0.	0.

35	-.107	-.000	-.171	.01082	1.5226E-03	-117.36	2.6716E-09	-119.86	7.00E-04	1.35E-03	-1.29E-09	-2.34E-09	0.	0.
36	-.078	-.134	.114	.00926	1.2522E-03	-.09	4.7391E-03	-2.85	1.08E-03	-1.78E-06	-6.26E-04	1.03E-06	4.73E-03	-2.35E-06
37	-.078	-.134	0.000	.00926	1.0176E-03	-4.94	5.1150E-03	-32.80	8.78E-04	-7.59E-05	-5.07E-04	4.38E-05	4.30E-03	-2.77E-03
38	-.078	-.134	-.114	.00926	1.4461E-03	-7.21	4.1035E-03	-55.66	1.28E-03	-1.61E-04	-7.37E-04	9.32E-05	2.31E-03	-3.39E-03
39	.000	-.155	.114	.00926	1.3423E-03	20.12	4.9566E-03	12.21	1.26E-03	4.62E-04	6.91E-13	2.53E-13	4.84E-03	1.05E-03
40	.000	-.155	0.000	.00926	1.1687E-03	21.77	4.9901E-03	-32.42	1.09E-03	4.33E-04	5.95E-13	2.38E-13	4.21E-03	-2.68E-03
41	.000	-.155	-.114	.00926	1.5473E-03	12.19	4.2640E-03	-54.07	1.51E-03	3.27E-04	8.30E-13	1.79E-13	2.50E-03	-3.45E-03
42	-.053	-.093	.171	.01082	1.4481E-02	-108.64	1.3284E-03	2.46	3.46E-03	6.91E-03	3.34E-03	1.19E-02	0.	0.
43	-.053	-.093	-.171	.01082	1.6060E-03	-93.33	1.4794E-03	179.58	1.33E-03	7.70E-04	-6.59E-04	1.41E-03	0.	0.
44	.078	-.134	.114	.00926	1.5629E-03	54.20	4.4093E-03	-8.25	7.92E-04	1.10E-03	4.57E-04	6.34E-04	4.36E-03	-6.33E-04
45	.078	-.134	0.000	.00926	1.5414E-03	69.22	4.7713E-03	-32.39	4.74E-04	1.25E-03	2.73E-04	7.21E-04	4.11E-03	-2.61E-03
46	.078	-.134	-.114	.00926	1.6650E-03	46.10	3.7035E-03	-48.41	1.00E-03	1.04E-03	5.77E-04	6.00E-04	2.46E-03	-2.77E-03
47	.134	-.078	.114	.00926	1.8492E-03	90.05	5.2792E-03	-11.18	-8.30E-07	9.25E-04	-1.44E-06	1.60E-03	5.18E-03	-1.02E-03
48	.134	-.078	0.000	.00926	3.9241E-03	108.44	4.7337E-03	-32.10	-6.21E-04	1.86E-03	-1.04E-03	3.22E-03	4.01E-03	-2.52E-03
49	.134	-.078	-.114	.00926	1.7239E-03	86.21	2.3165E-03	-31.90	5.69E-05	8.60E-04	9.86E-05	1.49E-03	1.97E-03	-1.22E-03
50	.053	-.093	.171	.01082	1.3417E-02	-104.86	1.4844E-03	36.96	-7.45E-04	-5.90E-03	3.66E-03	1.20E-02	0.	0.
51	.053	-.093	-.171	.01082	1.6588E-03	-49.57	1.5667E-03	-148.46	1.69E-03	7.85E-05	-2.64E-04	1.50E-03	0.	0.
52	.026	.026	.171	.00271	3.5238E-02	-97.22	3.6012E-02	-98.96	-4.43E-03	-3.50E-02	-5.61E-03	-3.56E-02	0.	0.
53	-.026	.026	.171	.00271	3.6415E-02	79.20	3.5986E-02	-98.91	6.42E-03	3.58E-02	-5.58E-03	-3.56E-02	0.	0.
54	-.026	-.026	.171	.00271	3.6415E-02	79.20	3.5986E-02	81.09	6.42E-03	3.58E-02	5.58E-03	3.56E-02	0.	0.
55	.026	-.026	.171	.00271	3.5238E-02	-97.22	3.6012E-02	81.04	-4.43E-03	-3.50E-02	5.61E-03	3.56E-02	0.	0.
56	0.000	0.000	-.171	.01082	4.6263E-03	-12.22	1.6046E-09	37.95	4.52E-03	-9.79E-04	-1.27E-09	-9.87E-10	0.	0.

- - - POWER BUDGET - - -

```
INPUT POWER    = 5.5651E-04 WATTS
RADIATED POWER= 5.5651E-04 WATTS
STRUCTURE LOSS= 0.         WATTS
NETWORK LOSS  = 0.         WATTS
EFFICIENCY    = 100.00 PERCENT
```

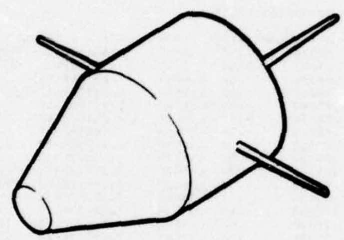
- - - RADIATION PATTERNS - - -

- - ANGLES - -		- POWER GAINS -			- - - POLARIZATION - - -			- - - E(THETA) - - -		- - - E(PHI) - - -	
THETA DEGREES	PHI DEGREES	VERT. DB	HOR. DB	TOTAL DB	AXIAL RATIO	TILT DEG.	SENSE	MAGNITUDE VOLTS/M	PHASE DEGREES	MAGNITUDE VOLTS/M	PHASE DEGREES
0.00	0.00	-8.78	-128.32	-8.78	.00000	-.00	LINEAR	1.03226E-01	1.99	1.08873E-07	160.97
5.00	0.00	-9.16	-127.99	-9.16	.00000	-.00	LINEAR	9.88070E-02	2.81	1.13000E-07	157.91
10.00	0.00	-9.86	-127.69	-9.86	.00000	-.00	LINEAR	9.11714E-02	2.66	1.17071E-07	154.87
15.00	0.00	-10.98	-127.40	-10.98	.00000	-.00	LINEAR	8.01192E-02	1.16	1.21067E-07	151.89
20.00	0.00	-12.69	-127.12	-12.69	.00000	-.00	LINEAR	6.58005E-02	-2.44	1.24964E-07	149.01
25.00	0.00	-15.28	-126.86	-15.28	.00000	-.00	LINEAR	4.88531E-02	-10.01	1.28734E-07	146.24
30.00	0.00	-19.20	-126.62	-19.20	.00000	-.00	LINEAR	3.11060E-02	-27.97	1.32347E-07	143.62
35.00	0.00	-22.86	-126.40	-22.86	.00000	-.00	LINEAR	2.04029E-02	-78.04	1.35774E-07	141.17
40.00	0.00	-19.12	-126.20	-19.12	.00000	.00	LINEAR	3.14011E-02	-132.16	1.38984E-07	138.90
45.00	0.00	-14.56	-126.01	-14.56	.00000	.00	LINEAR	5.30517E-02	-154.12	1.41949E-07	136.83
50.00	0.00	-11.32	-125.85	-11.32	.00000	.00	LINEAR	7.70493E-02	-166.25	1.44644E-07	134.96
55.00	0.00	-8.91	-125.71	-8.91	.00000	.00	LINEAR	1.01633E-01	-175.68	1.47050E-07	133.31
60.00	0.00	-7.02	-125.58	-7.02	.00000	.00	LINEAR	1.26366E-01	175.69	1.49151E-07	131.88
65.00	0.00	-5.46	-125.49	-5.46	.00000	.00	LINEAR	1.51240E-01	167.26	1.50935E-07	130.66
70.00	0.00	-4.13	-125.40	-4.13	.00000	.00	LINEAR	1.76407E-01	158.87	1.52397E-07	129.67
75.00	0.00	-2.95	-125.33	-2.95	.00000	.00	LINEAR	2.01988E-01	150.56	1.53533E-07	128.90
80.00	0.00	-1.90	-125.29	-1.90	.00000	.00	LINEAR	2.27919E-01	142.42	1.54341E-07	128.35
85.00	0.00	-.96	-125.26	-.96	.00000	.00	LINEAR	2.53857E-01	134.57	1.54823E-07	128.01
90.00	0.00	-.14	-125.25	-.14	.00000	.00	LINEAR	2.79144E-01	127.10	1.54978E-07	127.90
95.00	0.00	.57	-125.26	.57	.00000	.00	LINEAR	3.02850E-01	120.07	1.54806E-07	128.01
100.00	0.00	1.15	-125.29	1.15	.00000	.00	LINEAR	3.23865E-01	113.53	1.54308E-07	128.33
105.00	0.00	1.60	-125.33	1.60	.00000	.00	LINEAR	3.41012E-01	107.49	1.53483E-07	128.88
110.00	0.00	1.90	-125.40	1.90	.00000	.00	LINEAR	3.53162E-01	101.95	1.52332E-07	129.64
115.00	0.00	2.05	-125.49	2.05	.00000	.00	LINEAR	3.59340E-01	96.92	1.50855E-07	130.63
120.00	0.00	2.04	-125.59	2.04	.00000	.00	LINEAR	3.58811E-01	92.42	1.49057E-07	131.84
125.00	0.00	1.85	-125.71	1.85	.00000	.00	LINEAR	3.51141E-01	88.48	1.46943E-07	133.27
130.00	0.00	1.48	-125.86	1.48	.00000	.00	LINEAR	3.36233E-01	85.15	1.44526E-07	134.92
135.00	0.00	.89	-126.02	.89	.00000	.00	LINEAR	3.14356E-01	82.54	1.41820E-07	136.78
140.00	0.00	.08	-126.20	.08	.00000	.00	LINEAR	2.86150E-01	80.81	1.38847E-07	138.85
145.00	0.00	-1.01	-126.41	-1.01	.00000	.00	LINEAR	2.52651E-01	80.25	1.35630E-07	141.12
150.00	0.00	-2.39	-126.63	-2.39	.00000	.00	LINEAR	2.15341E-01	81.37	1.32199E-07	143.58
155.00	0.00	-4.13	-126.87	-4.13	.00000	.00	LINEAR	1.76320E-01	85.09	1.28583E-07	146.20
160.00	0.00	-6.21	-127.13	-6.21	.00000	.00	LINEAR	1.38785E-01	93.17	1.24814E-07	148.97
165.00	0.00	-8.37	-127.41	-8.37	.00000	.00	LINEAR	1.08185E-01	108.60	1.20921E-07	151.86
170.00	0.00	-9.67	-127.70	-9.67	.00000	.00	LINEAR	9.32114E-02	133.05	1.16932E-07	154.84
175.00	0.00	-9.08	-128.00	-9.08	.00000	.00	LINEAR	9.97322E-02	159.14	1.12870E-07	157.88
180.00	0.00	-7.37	-128.33	-7.37	.00000	.00	LINEAR	1.21475E-01	177.83	1.08756E-07	160.95
185.00	0.00	-5.61	-128.66	-5.61	.00000	.00	LINEAR	1.48638E-01	-170.31	1.04609E-07	164.02
190.00	0.00	-4.17	-129.02	-4.17	.00000	.00	LINEAR	1.75560E-01	-162.15	1.00446E-07	167.04
195.00	0.00	-3.05	-129.38	-3.05	.00000	.00	LINEAR	1.99710E-01	-155.65	9.62825E-08	170.00
200.00	0.00	-2.20	-129.77	-2.20	.00000	.00	LINEAR	2.20147E-01	-149.73	9.21393E-08	172.86
205.00	0.00	-1.57	-130.16	-1.57	.00000	.00	LINEAR	2.36811E-01	-143.80	8.80393E-08	175.60
210.00	0.00	-1.09	-130.57	-1.09	.00000	.00	LINEAR	2.50185E-01	-137.54	8.40104E-08	178.20
215.00	0.00	-.72	-130.98	-.72	.00000	.00	LINEAR	2.61061E-01	-130.81	8.00856E-08	-179.35
220.00	0.00	-.42	-131.40	-.42	.00000	.00	LINEAR	2.70321E-01	-123.62	7.63027E-08	-177.07
225.00	0.00	-.15	-131.82	-.15	.00000	.00	LINEAR	2.78717E-01	-116.05	7.27032E-08	-174.96
230.00	0.00	.09	-132.24	.09	.00000	.00	LINEAR	2.86693E-01	-108.25	6.93309E-08	-173.03
235.00	0.00	.32	-132.63	.32	.00000	.00	LINEAR	2.94244E-01	-100.41	6.62299E-08	-171.30
240.00	0.00	.52	-133.01	.52	.00000	.00	LINEAR	3.01132E-01	-92.67	6.34430E-08	-169.76
245.00	0.00	.68	-133.35	.68	.00000	.00	LINEAR	3.06542E-01	-85.17	6.10102E-08	-168.44
250.00	0.00	.77	-133.64	.77	.00000	.00	LINEAR	3.09849E-01	-77.97	5.89670E-08	-167.33
255.00	0.00	.78	-133.89	.78	.00000	-.00	LINEAR	3.10175E-01	-71.11	5.73433E-08	-166.45
260.00	0.00	.69	-134.07	.69	.00000	-.00	LINEAR	3.05966E-01	-64.60	5.61633E-08	-165.82
265.00	0.00	.48	-134.18	.48	.00000	-.00	LINEAR	2.99870E-01	-58.43	5.54444E-08	-165.43
270.00	0.00	.16	-134.22	.16	.00000	-.00	LINEAR	2.88850E-01	-52.59	5.51974E-08	-165.30
275.00	0.00	-.30	-134.18	-.30	.00000	-.00	LINEAR	2.74165E-01	-47.09	5.54263E-08	-165.44
280.00	0.00	-.88	-134.07	-.88	.00000	-.00	LINEAR	2.56364E-01	-41.91	5.61283E-08	-165.83
285.00	0.00	-1.59	-133.89	-1.59	.00000	-.00	LINEAR	2.36232E-01	-37.10	5.72938E-08	-166.47
290.00	0.00	-2.42	-133.65	-2.42	.00000	-.00	LINEAR	2.14731E-01	-32.66	5.89061E-08	-167.35
295.00	0.00	-3.35	-133.36	-3.35	.00000	-.00	LINEAR	1.92916E-01	-28.63	6.09420E-08	-168.46
300.00	0.00	-4.35	-133.02	-4.35	.00000	-.00	LINEAR	1.71954E-01	-25.06	6.33720E-08	-169.79
305.00	0.00	-5.39	-132.64	-5.39	.00000	-.00	LINEAR	1.52534E-01	-21.98	6.61607E-08	-171.33
310.00	0.00	-6.40	-132.24	-6.40	.00000	-.00	LINEAR	1.35788E-01	-19.38	6.92682E-08	-173.06
315.00	0.00	-7.31	-131.83	-7.31	.00000	-.00	LINEAR	1.22212E-01	-17.19	7.26513E-08	-174.99
320.00	0.00	-8.06	-131.41	-8.06	.00000	-.00	LINEAR	1.12106E-01	-15.26	7.62653E-08	-177.10
325.00	0.00	-8.59	-130.99	-8.59	.00000	-.00	LINEAR	1.05451E-01	-13.38	8.00657E-08	-179.38
330.00	0.00	-8.89	-130.57	-8.89	.00000	-.00	LINEAR	1.01914E-01	-11.36	8.40102E-08	178.18
335.00	0.00	-8.98	-130.16	-8.98	.00000	-.00	LINEAR	1.00905E-01	-9.08	8.80602E-08	175.58
340.00	0.00	-8.91	-129.76	-8.91	.00000	-.00	LINEAR	1.01637E-01	-6.60	9.21817E-08	172.84
345.00	0.00	-8.78	-129.38	-8.78	.00000	-.00	LINEAR	1.03200E-01	-4.06	9.63459E-08	169.99
350.00	0.00	-8.66	-129.01	-8.66	.00000	-.00	LINEAR	1.04618E-01	-1.64	1.00529E-07	167.04
355.00	0.00	-8.64	-128.66	-8.64	.00000	-.00	LINEAR	1.04922E-01	.44	1.04711E-07	164.02
360.00	0.00	-8.78	-128.32	-8.78	.00000	-.00	LINEAR	1.03228E-01	1.99	1.08873E-07	160.97

***** DATA CARD NO. 4 EN -0 -0 -0 -0 -0. -0. -0. -0. -0. -0.

3.0 FORMULATION

The theory behind the basic AMP code which uses thin wires in modeling structures is outlined in detail in the Engineering Manual (reference 2). Of main interest here is the modification to the AMP code which allows for the modeling of a generally shaped voluminous structure by means of surface patches as an alternative to wire grid modeling. An example of the general type of structure being considered is illustrated below; in this case thin wire appendages are connected to a conducting voluminous structure.



As described in the Engineering Manual, the electric field integral equation (EFIE) specialized to thin conducting wires is being used in the basic AMP. Though the EFIE could be used for the voluminous structure as well, the magnetic field integral equation (MFIE) is generally more attractive for this case[4]; in particular this is true for structures having a large smooth surface. Therefore, both the MFIE and the EFIE are being used in the modified program to obtain the currents for structures of the type illustrated above.

Using notation which is similar to that used in the Engineering manual, the EFIE and the MFIE[4] can be written respectively

$$-\hat{n}(\overline{r}_0) \times \overline{E}^I(\overline{r}_0) = -\frac{i\eta_0}{4\pi k}\,\hat{n}(\overline{r}_0) \times \oint_S \left[\overline{J}_s(\overline{r})\,k^2 + (\overline{J}_s(\overline{r}) \cdot \overline{\nabla})\,\overline{\nabla}\right] g(\overline{r},\overline{r}_0)\,dA \qquad (1)$$

and

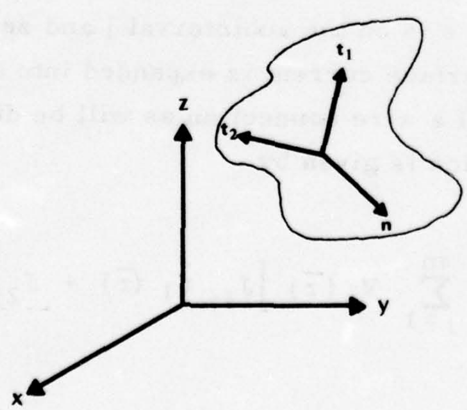
$$-\hat{n}(\overline{r}_0) \times \overline{H}^I(\overline{r}_0) = -1/2\,\overline{J}_s(\overline{r}_0) + \frac{1}{4\pi}\,\hat{n}(\overline{r}_0) \times \oint_S \overline{J}_s(\overline{r}) \times \overline{\nabla} g(\overline{r},\overline{r}_0)\,dA \qquad (2)$$

where the integration is carried out over the surface enclosing the entire body excluding the singularity as indicated by the principal value integral sign. As discussed above, for our application the EFIE will be enforced only on wire portions of the structure and the MFIE is enforced only on the large surface portions of the structure. Thus $\overline{r}_o$ which locates the observation point on the surface is restricted to wires in equation (1), and in equation (2) $\overline{r}_o$ is restricted to the large surface areas; together, the two equations account for the entire structure.

With the thin wire approximation included equation (1) becomes

$$-\hat{s}_o \cdot \overline{E}^I(\overline{r}_o) = - \frac{i\eta_o}{4\pi k} \int_L I(s) \left[\hat{s} \cdot \hat{s}_o k^2 - \frac{\partial^2}{\partial s\, \partial s_o}\right] g(\overline{r}, \overline{r}_o)\, ds - \frac{i\eta_o}{4\pi k}\ \hat{s}_o \cdot \int_{S_1} \left[\overline{J}_s(\overline{r})\, k^2 + (\overline{J}_s(\overline{r}) \cdot \overline{\nabla})\, \overline{\nabla}\right] g(\overline{r}, \overline{r}_o)\, dA \qquad (3)$$

where the integration over the surface in the second integral now of course excludes surface portions covering the wires. In order to reduce the vector equation (2) to two scalar components, a local coordinate system is defined as shown in the illustration such that the unit vectors $\hat{t}_1$ and $\hat{t}_2$ are orthogonal vectors tangent to the surface and $\hat{n}$ is the normal vector as before. Now using the identity $\overline{u} \cdot (\overline{v} \times \overline{w}) = (\overline{u} \times \overline{v}) \cdot \overline{w}$



and noting the fact that $\hat{t}_1 \times \hat{n} = -\hat{t}_2$ and $\hat{t}_2 \times \hat{n} = \hat{t}_1$, equation (2) with the thin wire approximation included becomes

$$\hat{t}_2(\overline{r}_o) \cdot \overline{H}^I(\overline{r}_o) = -\frac{1}{4\pi}\hat{t}_2(\overline{r}_o) \cdot \int_L I(s)\left(\hat{s} \times \overline{\nabla} g(\overline{r}, \overline{r}_o)\right) ds$$

$$-\frac{1}{2}\hat{t}_1(\overline{r}_o) \cdot \overline{J}_s(\overline{r}_o) - \frac{1}{4\pi}\hat{t}_2(\overline{r}_o) \cdot \fint_{S_1} \overline{J}_s(\overline{r}) \times \overline{\nabla} g(\overline{r}, \overline{r}_o)\, dA \quad (4a)$$

and

$$-\hat{t}_1(\overline{r}_o) \cdot \overline{H}^I(\overline{r}_o) = \frac{1}{4\pi}\hat{t}_1(\overline{r}_o) \cdot \int_L I(s)\left(\overline{s} \times \overline{\nabla} g(\overline{r}, \overline{r}_o)\right) ds$$

$$-\frac{1}{2}\hat{t}_2(\overline{r}_o) \cdot \overline{J}_s(\overline{r}_o) + \frac{1}{4\pi}\hat{t}_1(\overline{r}_o) \cdot \fint_{S_1} \overline{J}_s(\overline{r}) \times \overline{\nabla} g(\overline{r}, \overline{r}_o)\, dA \quad (4b)$$

These two components suffice since there is no normal component of equation (2).

The method of collocation as outlined in the Engineering manual is used to reduce the equations (3) and (4) to a system of linear equations. As before, the wire current is expanded into a set of functions having constant, sine and cosine terms, i.e.

$$I(s) = \sum_{j=1}^{N} U_j(s)\left[A_j + B_j \sin k(s - s_j) + C_j \cos k(s - sj)\right] \quad (5)$$

where $U_j(s)$ is 1 when s is on the subinterval j and zero otherwise. On the other hand, the surface current is expanded into a set of pulse functions except in the region of a wire connection as will be discussed later. The pulse function expansion is given by

$$\overline{J}_s(\overline{r}) = \sum_{j=1}^{m} V_j(\overline{r})\left[J_{1j}\,\hat{t}_1(\overline{r}) + J_{2j}\,\hat{t}_2(\overline{r})\right] \quad (6)$$

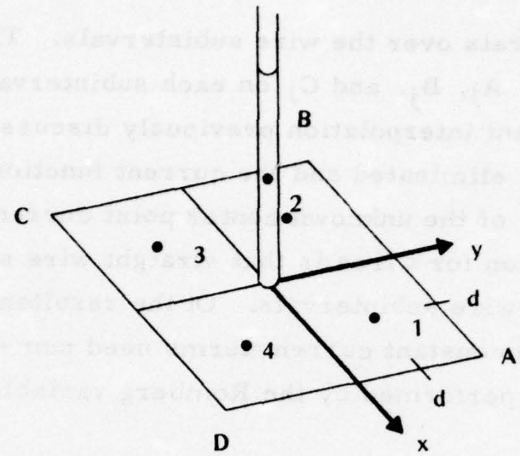
where $V_j(\overline{r})$ is 1 for points in the jth subsection and zero elsewhere. These expansions are substituted into equations (3) and (4). The wire integrals in equations (3) and (4) can then be written in terms of a summation of integrals over the wire subintervals. There are now three unknown quantities, A_j, B_j, and C_j on each subinterval; however, through the use of the current interpolation previously discussed [2], two of these unknowns are eliminated and the current function is then expressed in terms of the unknown center point currents on each interval. A final approximation for wires is that straight wire segments are used to approximate the wire subintervals. Of the resultant integrals, only those related to the constant current terms need numerical integration. This integration is performed by the Romberg variable-interval width technique [2].

The terms involving surface integrations in equations (3) and (4) are handled in a simpler manner than the wire terms. Since pulse functions are used for the current expansion, no current interpolation is necessary. The two unknown current components, J_{1j} and J_{2j}, on each surface subsection will be accounted for by enforcing the two equations, 4(a) and 4(b), on the center of each subsection. The surface subsections are approximated by flat surface patches and the resulting integrals are evaluated in one step; that is, the value of the integral is equal to the product of the kernel at the center of the patch and the patch area. No special consideration is necessary for the case of the source and observation point on the same patch as happens in equation (4) since these terms are identically equal to zero. Note that for a flat surface, the resultant vector of the surface integral in equation (4) is normal to the surface; thus, when dotted with surface tangent vectors, these terms are identically equal to zero.

More accurate treatment of the surface integral in equation (3) is necessary in the region of a wire connection. The treatment used is quite similar to that presented by Albertsen et al [5]. After Albertsen, the surface current density, $\overline{J}_s$, around the connection point should meet the requirement

$$\overline{\nabla}_s \cdot \overline{J}_s (x, y) = J_0 (x, y) + I_0 \delta (x, y)$$

where the coordinates are defined in the illustration below, $\overline{\nabla}_s \cdot$ denotes surface divergence, $J_0 (x, y)$ is a continuous function in the region ABCD, and I_0 is the wire current flowing onto the surface. One expansion which



meets this requirement is

$$\overline{J}_s (x, y) = I_0 \overline{f} (x, y) + \sum_{j=1}^{4} g_j (x, y) (\overline{J}_j - I_0 \overline{f}_j) \tag{7}$$

where

$$\overline{f} (x, y) = \frac{x\hat{x} + y\hat{y}}{2\pi (x^2 + y^2)}$$

$$\overline{J}_j = \overline{J}_s (x_j, y_j)$$

$$\overline{f}_j = \overline{f} (x_j, y_j)$$

and the interpolation functions $g_j (x, y)$ are chosen such that: $g_j (x, y)$ is differentiable on ABCD; $g_j (x_i, y_i) = \delta_{ij}$; and $\sum_{j=1}^{4} g_j (x, y) = 1$. The specific functions used in AMP are the following:

$$g_1 (x, y) = \frac{1}{4d^2} (d+x)(d+y) \qquad g_2 (x, y) = \frac{1}{4d^2} (d-x)(d+y)$$

$$g_3 (x, y) = \frac{1}{4d^2} (d-x)(d-y) \qquad g_4 (x, y) = \frac{1}{4d^2} (d+x)(d-y)$$

When computing the field in equation (3) on the attachment wire segment which is due to the surface current, the expansion for the current given in equation (7) is used in place of the pulse function expansion over the flat square region ABCD. The current flowing onto the surface is, to a good approximation, equal to the current at the center of the attachment segment; therefore, I_o in (7) is set equal to the segment centerpoint current. The integration over the region is performed numerically using the rectangular rule.

At this point the equations can be reduced to matrix form. The solution of this matrix equation for the currents is obtained using the Gauss-Doolittle factorization method as described in the Engineering manual. In addition, structure symmetry can again be used to greatly reduce the time taken in the matrix solution; the techniques employed are the same as those for wires alone. Once the structure currents have been obtained, the far electric field is readily calculated by summing the electric field from the wire and surface portions of the structure. The expression for the field can be written

$$\overline{E}(\overline{r}_o) = \frac{ik\eta_o}{4\pi} \frac{e^{-ikr_o}}{r_o} \left[\int_L \left[(\hat{k} \cdot \overline{I}(s))\hat{k} - \overline{I}(s) \right] e^{i\overline{k} \cdot \overline{r}} ds + \int_{S_1} \left[(\hat{k} \cdot \overline{J}_s(\overline{r}))\hat{k} - \overline{J}_s(\overline{r}) \right] e^{i\overline{k} \cdot \overline{r}} dA \right] \tag{8}$$

where $\overline{r}_o$ is the vector from the origin of coordinates to the observation point, $\hat{k}$ is in the direction of $\overline{r}_o$, and $k = 2\pi/\lambda$. The evaluation of the wire term in equation (8) is outlined in the Systems manual (3), and the evaluation of the surface term is by straight forward rectangular rule integration.

To demonstrate the validity of the approach taken for the solution of the surface-wire structure problem, three comparisons of computed and experimental radiated field patterns are presented in Figures 5, 6, and 7.

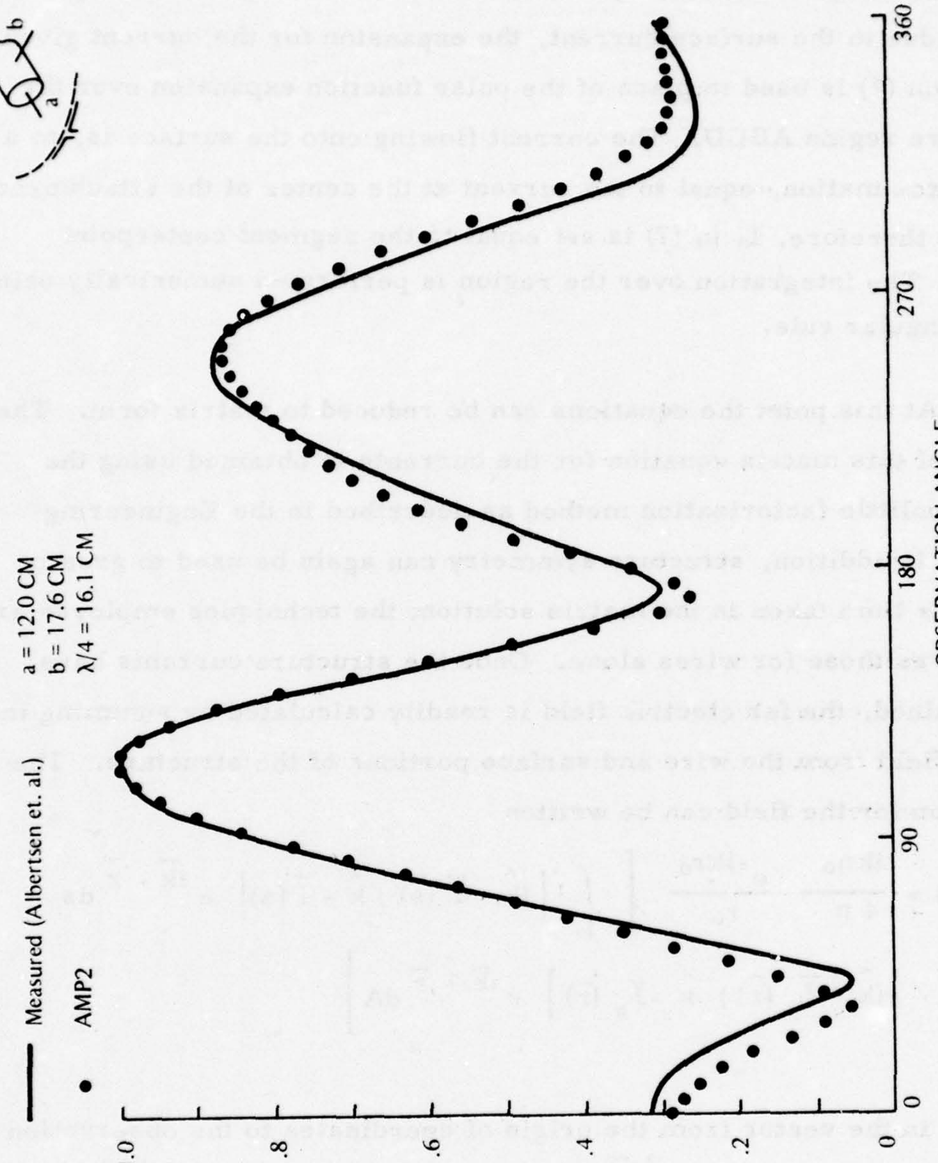


FIGURE 5
EXPERIMENTAL/NUMERICAL RADIATION PATTERN OF CYLINDER WITH ATTACHED WIRES

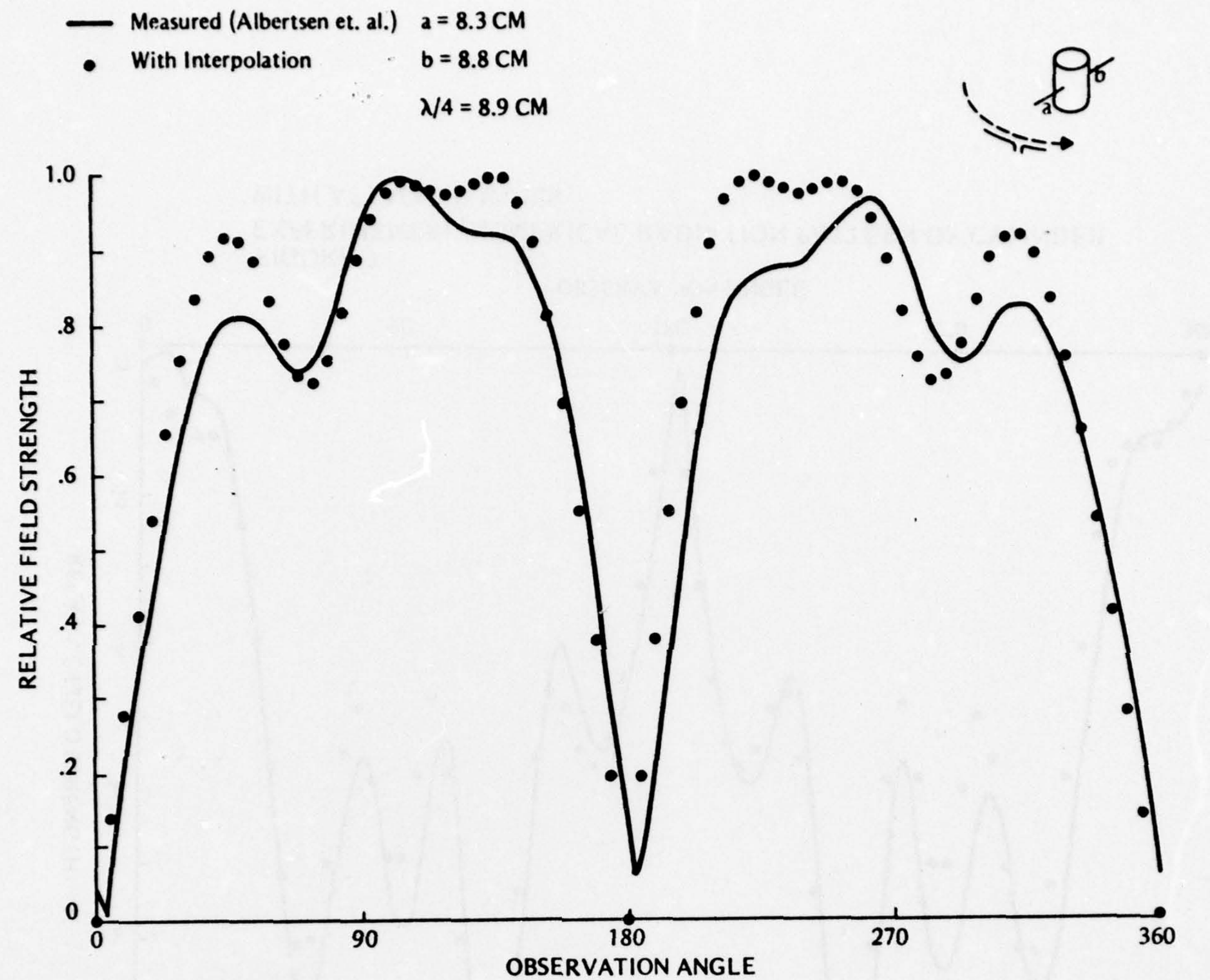


FIGURE 6
EXPERIMENTAL/NUMERICAL RADIATION PATTERN
WITH ATTACHED WIRES

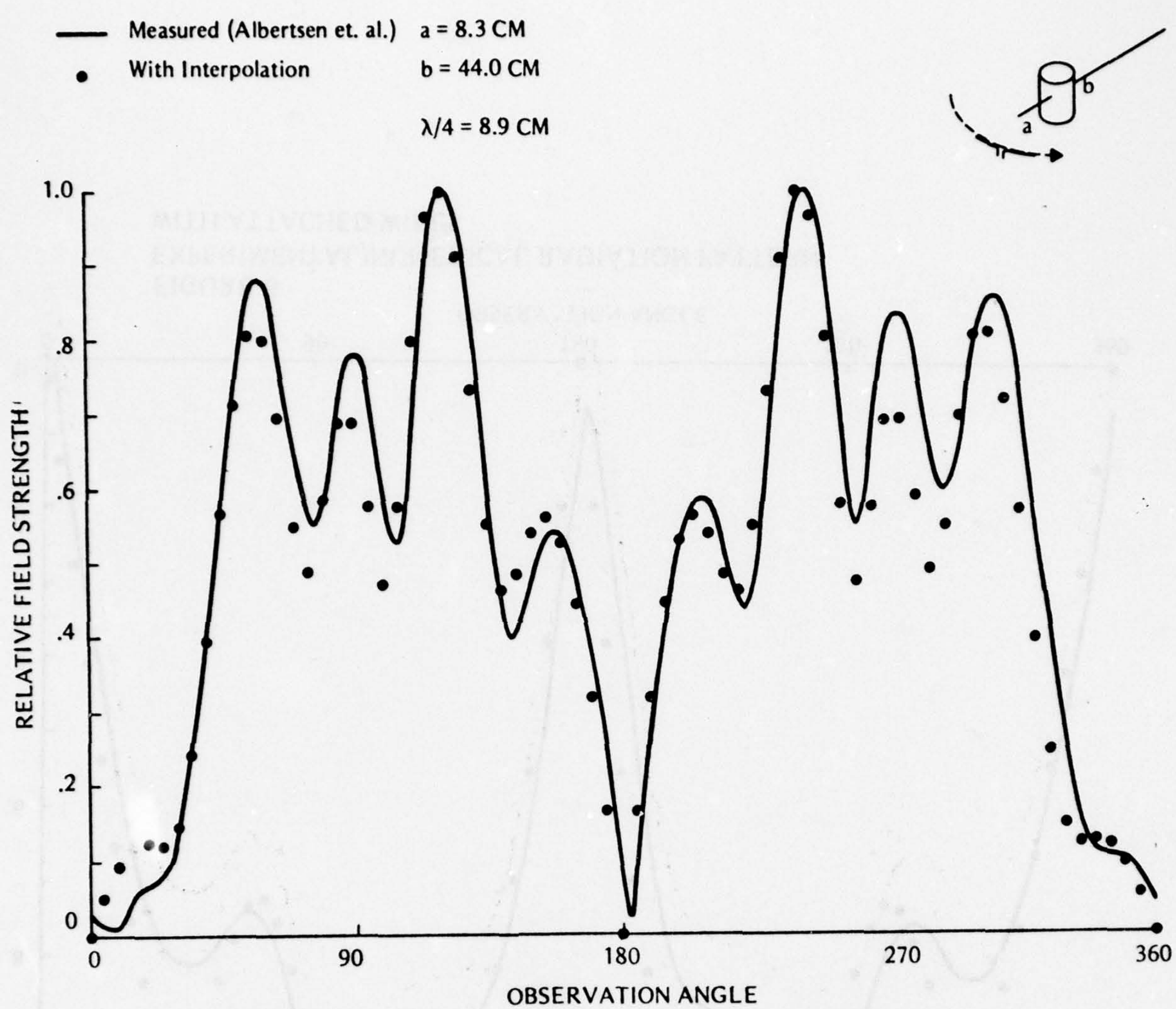


FIGURE 7
EXPERIMENTAL/NUMERICAL RADIATION PATTERN OF CYLINDER WITH ATTACHED WIRES

The basic structure is a cylinder whose length is 22 cm and whose diameter is 20 cm. To this cylinder, two wires are attached in various locations as illustrated in the upper right hand corner of the figures. Wire a is driven and wire b is passive in each case. The experimental measurements were made at the Univeristy of Denmark by Albertsen et al[5]. The numerical computations were made using the cylinder model illustrated in figure 3(c); furthermore, figure 3(d) illustrates the specific wire attachment case whose results appear in Figure 5. No special consideration has been given to the edges of the cylinder, but in view of the results shown in the three figures here and results for other configurations obtained by Albertsen, this is apparently not of significant practical importance. On the other hand, increased patch density near an edge has yielded more accurate results in certain instances. It should be pointed out, however, that the MFIE is not valid for field points directly on ideally sharp edges; moreover, in this formulation wires cannot be attached directly on an edge.

3.1 APPROXIMATE MATRIX ELEMENTS

When wire segments in a structure are distant from an observation point with respect to wavelength, simple expressions can be used to obtain accurate values for the fields. This fact can be used to substantially reduce the time required in calculating the corresponding interaction matrix elements. The following expressions are used in the AMP2 code when segment-observation point separation permits:

$$E_r(\bar{r}_o) = \frac{I\ell}{2\pi} e^{-ikr_o} \left(\frac{\eta}{r_o^2} + \frac{1}{i\omega\epsilon r_o^3} \right) \cos\theta \tag{9a}$$

$$E_\theta(\bar{r}_o) = \frac{I\ell}{4\pi} e^{-ikr_o} \left(\frac{i\omega\mu}{r_o} + \frac{\eta}{r_o^2} + \frac{1}{i\omega\epsilon r_o^3} \right) \sin\theta \tag{9b}$$

$$H_\varphi(\bar{r}_o) = \frac{I\ell}{4\pi} e^{-ikr_o} \left(\frac{ik}{r_o} + \frac{1}{r_o^2} \right) \sin\theta \tag{9c}$$

These are the fields of an incremental dipole of moment $I\ell$ located at the origin of a standard spherical coordinate system and oriented in the z direction[6]. At sufficient distances equation (9) is used for the field of a segment where ℓ is set equal to the segment length and I is set equal to the center point current. Thus, these expressions are the same as would be obtained using a pulse function current expansion and one step integration.

This approximation has been found to yield good results for separation distances as small as .25 to .2 wavelengths. Table 1 shows the accuracy obtained for a particular structure, a 2λ dipole, for various segmentations and for various separation distances for which the expressions in equation (9) were used. The KH parameter in the table specifies the distance criterion in wavelengths where separations greater than the criterion use the expressions in (9) and separations less than the criterion use accurate integration over the segment. The column on the left hand side of the table shows the number of segments away from the self segment which are integrated over. For this example it can be seen that the impedance accuracy remains within a few percent for a KH down to .21 wavelengths. It should be pointed out, however, that due to the quantized nature of the problem a KH parameter slightly less than .2 wavelengths will cause an abrupt change to integration over one fewer segments. For the case of .2 λ segment lengths, this means integration for the self term only and the results are poor. This problem can be avoided by keeping the KH parameter larger than the longest segment. It should also be pointed out that the minimum value for KH seems to depend to some extent on the structure size, type, segmentation, and excitation. Values of KH up to .5 λ have been necessary to obtain only a few percent error for some structures with very asymmetric feeds. No exact guidelines have been established; therefore, it is probably best to experiment with any given class of problems if a minimum value of KH is sought. The default value for the KH parameter in the AMP code is one wavelength.

NUMBER OF SEGMENTS INCLUDED	STRUCTURE SEGMENT LENGTHS		
	0.2	0.1	0.05
0	KH PARAMETER .01	.01	.01
	% ERROR REAL, IMAG. 47.2, 53.	77.3, 135.	97.4, 170.4
1	.21	.11	.06
	2.2, .62	12.4, 12.4	21.4, 190
2	.41	.21	.11
	.068, .015	1.3, 2.4	.12, 31.5
3	.61	.31	.16
	.35, .015	.09, .09	.23, 9.5
4	.81	.41	.21
	.19, .33	.028, .30	.13, 3.
6	1.21	.61	.31
	.06, .003	.035, .12	.01, .23
8	1.65	.81	.41
	.02, .022	.09, .19	.037, .041

Table 1

PER CENT ERROR OF THE INPUT IMPEDANCE OF A 2λ DIPOLE USING PARTIAL INTEGRATION AS COMPARED TO COMPLETE INTEGRATION

4.0 DESCRIPTION OF THE COMPUTER CODE

In this section of the manual, the details of the additions and modifications to the AMP computer code will be discussed routine by routine. Only those routines which have been modified in some way other than the simple change of a common block or are entirely new will be discussed. A description of all other routines can be found in reference 3. The following list denotes those routines which have been modified and those which are new:

Modified	New
MAIN	APRXE
CABC	APRXH
CATLOG	FFLDS
CHKPRT	GH
CMSET	HFK
CONECT	HFLD
CONVRT	HINTG
DATAGN	HMAT
ETMNS	HWMAT
FACTR	MATFIL
FACTRS	PATCH
FFLD	PCINT
ISEGNO	UDOTES
JMELS	UNERE
LFACTR	
MOVE	
NETWK	
REFLC	
SOLVES	
WIRE	

The discussion of each of these routines follows these introductory remarks, and the discussions are arranged alphabetically by routine name for the entire group of routines.

In addition to the modified routines listed above, other routines have changed simply because common/DATA/and common/RESTRT/have been changed. Common/DATA/now acts as the geometry data storage location for both wire segments and surface patches; therefore, the total number of patches, M, and the number of patches in a symmetric section, MP, have been added to the variable list. Furthermore, in each array the wire segment data is stored from the beginning of the array sequentially to N, the total number of segments, while the patch data is stored in reverse sequential order starting at the last location in the array and proceeding for M locations. Thus the length of each array must be known, and this quantity, LD, is also included in the new common/DATA/. This storage of the segment and patch data in the same arrays provides the maximum program flexibility in a given amount of storage, but since variable names were chosen for the wire case, the variable names for the surface patch case are not necessarily mnemonic. The use of the arrays for surface patches is as follows:

X, Y, Z	= arrays containing the x, y, z coordinates of the patch centers in wavelengths
SI	= array containing t_{1x} for each patch
BI	= array containing patch areas in square wavelengths
ALP, BET	= arrays containing t_{1y} and t_{1z} respectively for each patch
ICON1, ICON2, ITAG	= arrays containing the x, y, z components of $\hat{t}_2$ respectively for each patch.

Most routines involved in surface calculations equivalence these variables to others which have more meaningful names.

The common block/ANGL/is also used to store patch data from the top of array SALP, sequentially downward. The quantity stored for a patch is +1 if ($\hat{t}_1$, $\hat{t}_2$, $\hat{n}$) for that patch form a right hand coordinate system and -1 if ($\hat{t}_1$, $\hat{t}_2$, $\hat{n}$) form a left hand coordinate system.

The common block/RESTRT/has been expanded to contain variables associated with the precautionary file dumping. The variables IDUMP, TMDUM, and EXTIM have been added. The variables are defined as follows:

IDUMP	= file dump flag, = 0 (default) no dumping, = 1 for dumping
TMDUM	= time interval in seconds between file dumps
EXTIM	= clock time in seconds at execution start

At this point we turn to the discussion of individual routines.

APRXE

PURPOSE: to compute the electric field due to an infinitesimal current element as the wire to wire interaction matrix element for large separation distances.

METHOD: The electric field due to a current element with current I and length Δ oriented along the z axis at the origin of a spherical coordinate system is

$$\bar{E} = \bar{E}' \frac{I}{\lambda}$$

$$\bar{E}' = E'_r \hat{r} + E'_\theta \hat{\theta}$$

where

$$E'_r = \frac{(\Delta/\lambda)\eta}{2\pi} e^{-i2\pi R} \left(\frac{1}{R^2} - i \frac{1}{2\pi} \frac{1}{R^3} \right) \cos\theta$$

$$E'_\theta = \frac{(\Delta/\lambda)\eta}{4\pi} e^{-i2\pi R} \left(\frac{i2\pi}{R} + \frac{1}{R^2} - i \frac{1}{2\pi} \frac{1}{R^3} \right) \sin\theta$$

$$\eta = \sqrt{\mu/\epsilon}$$

R is the distance from the current element to the observation point divided by wavelength (λ) and θ is the angle from the z axis to the vector to the observation point, $\hat{r}$. The component of $\bar{E}'$ parallel to the observation segment is the matrix element.

In the code, E'_r and E'_θ are computed at AE14 and AE15 respectively. The r and θ components are converted to z and ρ components in cylindrical coordinates at AE16 and AE17. At AE18 GN is called to modify the field for reflection from the ground if the case being computed is the field of the image of a segment in a ground plane (indicated by IP=2). Finally the matrix element is computed at AE19.

SYMBOL DICTIONARY:

A0	= $\cos\theta$
A1	= $\sin\theta$
C1	= $\exp(-i2\pi R)$
DIJ	= dot product of the unit vectors in the directions of the source and observation segments
DIR	= dot product of the unit vector in the direction of the observation segment with the ρ unit vector in the cylindrical coordinate system.
EP	= ρ component of $\overline{E}'$
EPE	= array for gaining access to the real and imaginary parts of EP
ER	= E'_r
ET	= E'_θ
ETA	= η
ETI	= imaginary part the matrix element
ETR	= real part of the matrix element
EZ	= z component of $\overline{E}'$
EZE	= array for gaining access to the real and imaginary parts of EZ
IP	= flag to indicate (if equal to 2) that field being computed is reflected in ground plane
PI2	= 2π
R	= distance from source to observation segment (in wavelengths)
RH	= ρ coordinate of the observation segment in a cylindrical coordinate system with origin at the center of the source segment and z axis in the direction of the source segment (in wavelengths).
RKH1	= $2\pi R$
S	= Δ/λ
ZP	= z coordinate of observation segment (see RH)

APRXH

PURPOSE: to compute the magnetic field due to an infinitesimal current element as the wire to patch interaction matrix element for large separation distances.

METHOD: The magnetic field due to a current element with current I and length Δ oriented along the z axis at the origin of a spherical coordinate system is

$$\overline{H} = \overline{H}' \frac{I}{\lambda}, \quad \overline{H}' = H'_{\varphi}\, \hat{\varphi}$$

$$H'_{\varphi} = \frac{(\Delta/\lambda)}{4\pi} e^{-j2\pi R} \left(\frac{1}{R^2} + i \frac{2\pi}{R} \right) \sin\theta$$

$$\eta = \sqrt{\mu/\epsilon}$$

The matrix elements for the two scalar components of the MFIE are $\mp\, \hat{t}_1 \cdot \overline{H}'$ where the upper sign applies for a patch where $(\hat{t}_1, \hat{t}_2, \hat{\eta})$ *form* a right hand system (normal case) and the lower sign applies for a left hand system (patch reflected in a symmetry plane). $\overline{H}'$ is computed at AH17 and the matrix element contributions at AH22 and AH23. The contributions are accumulated in TWHR and TWHI so that when a ground is present the routine can be called first for the direct field and then for the ground reflected field with the fields summed.

SYMBOL DICTIONARY:

H	$= H'_{\varphi}$
ILC	= location of the patch coordinate data in the common/DATA/arrays
K	= do loop parameter
PDT	$= \pm\, \hat{\varphi} \cdot (\hat{t}_1$ or $\hat{t}_2)$ upper sign is used when $(\hat{t}_1, \hat{t}_2, \hat{\eta})$ form right hand system
PX	= x component of $\hat{\varphi}$

PY	= y component of $\hat{\varphi}$
PZ	= z component of $\hat{\varphi}$
R	= distance from source segment to observation patch (in wavelengths)
RFL	$= \begin{cases} +1 & \text{for direct field of segment} \\ -1 & \text{for field reflected in ground} \end{cases}$
RK	$= 2\pi R$
R2	$= R^2$
S	$= \Delta/\lambda$
ST	$= \sin\theta$
TPI	$= 2\pi$
TWHI	= imaginary part of matrix elements
TWHR	= real part of matrix elements
T1X, T1Y, T1Z	= x, y and z components of $\hat{t}_1$
T2X, T2Y, T2Z	= x, y and z components of $\hat{t}_2$

CABC (modified)

PURPOSE: to compute the coefficients in the sinusoidal basis functions for the current on each segment, given current at the center of each segment. Also the patch currents are converted from two surface vector components to x, y and z components.

MODIFICATION: The code added from CB86 to CB101 converts the patch currents from the two components

$$\overline{J} = J_1 \hat{t}_1 + J_2 \hat{t}_2$$

to the three components

$$\overline{J} = J_x \hat{x} + J_y \hat{y} + J_z \hat{z}$$

The components J_x, J_y and J_z are stored in the array CUR in place of J_1 and J_2. If there are N segments the currents for patch i are stored as follows:

J_1 = CUR (N + 2i - 1) J_2 = CUR (N + 2i)

J_x = CUR (N + 3 i - 2) J_y = CUR (N + 3i - 1)

J_z = CUR (N + 3i)

Hence the conversion starts with the last patch and proceeds down in patch number to avoid writing over values of J_1 and J_2 before they are converted.

LOCAL SYMBOL DICTIONARY:

CLL	= J_2
CLO	= J_1
CUR	= array containing current values
JCO1	= N + 2i -1
JCO2	= N + 3i -2
K	= location of patch data in data arrays
LD	= dimensioned length of data arrays
M	= total number of patches

$$\left.\begin{array}{l} \text{T1X} \\ \text{T1Y} \\ \text{T1Z} \end{array}\right\} = \text{x, y and z components of } \hat{t}_1$$

$$\left.\begin{array}{l} \text{T2X} \\ \text{T2Y} \\ \text{T2Z} \end{array}\right\} = \text{x, y and z components of } \hat{t}_2$$

CATLOG (modified)

PURPOSE: to write the information contained on files 11-16 onto file 17. File 17 can then be used to restart the AMP program at the point where the files were dumped.

MODIFICATIONS: In AMP, the program execution was stopped after the files were dumped. The routine has been modified so that execution is not stopped for the case of the precautionary file dumps, rather control is returned to the calling program. The statements CT56 and CT57 replace the STOP statement.

Other small modifications entail changes in variable names in the common block RESTRT. The new common block variable names are equivalenced to the variable names used in the routine.

CHKPRT (modified)

PURPOSE: to check for interrupt during out of core matrix handling and to control automatic file dumping.

METHOD: This routine is called at convenient times for program interruption during out of core matrix filling and factorization. The current program run time is compared to an input quantity TMDUM, if the run time is greater, routine CATLOG is called to dump the scratch files onto file 17. The files are then repositioned in CHKPRT and control is returned to the calling program. The next dump will occur when the time since the last dump is greater than TMDUM. In case of machine failure, the program can be restarted from one of these file dumps.

The routine CHKPRT could be revised by the user to include other options relating to file dumping. Some of these options are discussed in the CHKPRT writeup in reference 3.

SYMBOL DICTIONARY:

EXTIM	= clock time in seconds at execution start
I	= loop index
ICK	= flag checking first call to CHKPRT
IDUMP	= flag indicating if auto dumping is desired
J	= number of backspaces required on certain files during repositioning
T	= running time in seconds when the dump is initiated
TMDUM	= minimum time between dumps in seconds, input quantity
T1	= clock time at start of checking period
T2	= current clock time in seconds

VARIABLES IN COMMON:

ICASE, IC1, IC2, IC3, IDUMP, NBLSYM, NPRES, NRES, TMDUM (other variables in common not referenced, see listing).

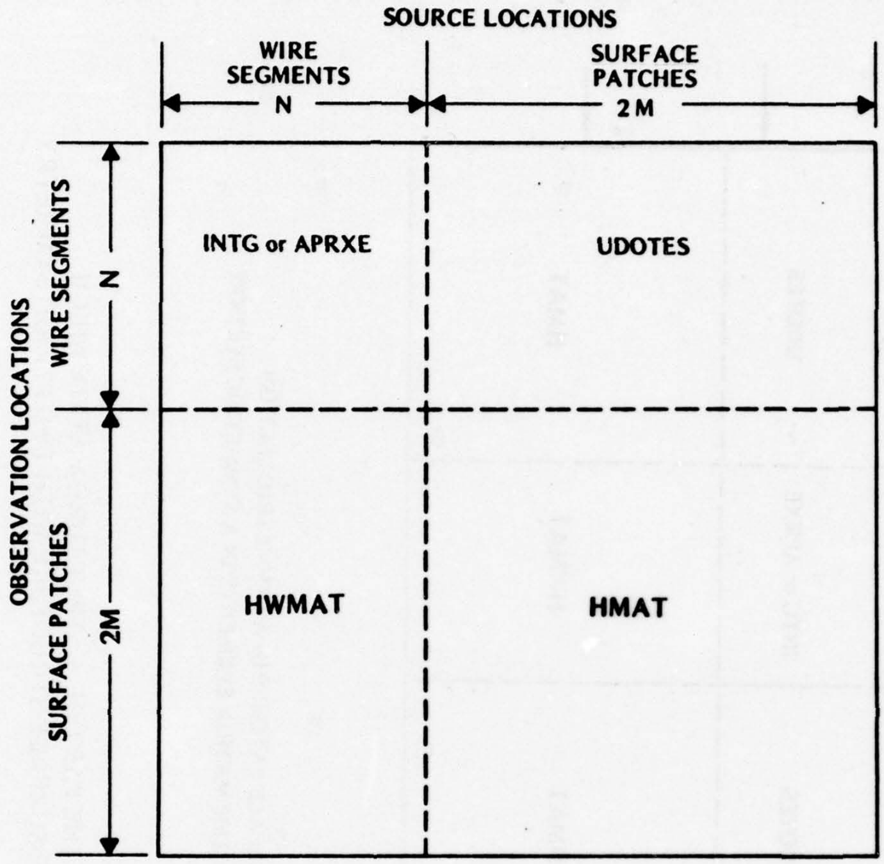
CMSET (modified)

PURPOSE: to control the filling of the structure matrix in the CM array.

METHOD: The structure matrix in the AMP2 code can be completely composed of wire elements or surface elements, or there can be a combination of wire and surface elements. The latter case will be discussed here since it includes the former cases. The layout of the structure matrix for an unsymmetric wire-surface structure is illustrated in figure 8. Filling of the matrix is arranged so that wire to wire interaction terms appear in the upper left hand box of the matrix and surface to surface interaction terms appear in the bottom right hand box of the matrix. The rectangular boxes at the other corners of the matrix contain the terms representing the interaction between wires and surfaces. The size of the boxes is governed by the number of wire segments, N, and the number of surface patches, M. CMSET calls other routines to actually calculate and fill the appropriate boxes in the matrix, and these routines have been indicated in figure 8 in their respective boxes. The routines INTG, APRXE, and HWMAT handle element calculations only, CMSET then calls MATFIL for placing the elements in the CM array.

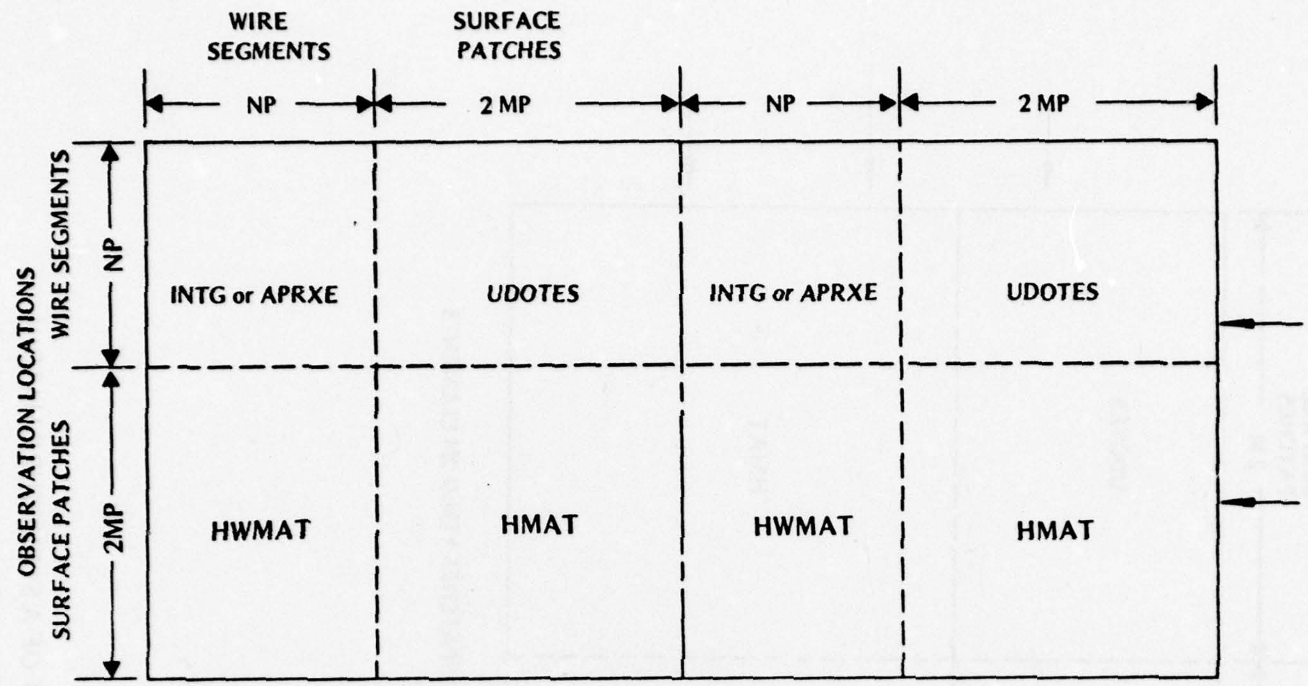
Figure 9 illustrates the matrix layout for the case of a wire-surface structure with two period symmetry. The complete matrix has the form

$$\begin{bmatrix} A & B \\ B & A \end{bmatrix}$$



NOTE: M SURFACE PATCHES YIELD 2M ELEMENTS

FIGURE 8
EXAMPLE LAYOUT OF A STRUCTURE MATRIX
FOR AN UNSYMMETRIC STRUCTURE WITH WIRES AND SURFACES



NOTE: MP SURFACE PATCHES IN A SYMMETRIC SECTION
YIELD 2 MP MATRIX ELEMENTS IN A SYMMETRIC SECTION

FIGURE 9
EXAMPLE LAYOUT OF THE PART OF A STRUCTURE MATRIX WHICH IS STORED FOR A WIRE-SURFACE STRUCTURE WITH TWO PERIOD SYMMETRY

however, the only part stored and shown in the figure are the unique blocks $\left[A \quad B\right]$. As in the unsymmetric case, the total number of unknowns is determined by the number of wire segments, N, plus twice the number of surface patches, M. The size of the submatrices A and B on the other hand are determined by the number of wire segments in a symmetric period, NP, plus twice the number of patches in a symmetric period, MP. The placement of the boxes representing the various types of interactions within the submatrices is shown in figure 9.

The filling of the matrix takes place in two major steps. All matrix elements representing wire segments as sources are filled first, and the elements related to patches as sources are filled second. The part of CMSET which handles wire sources is contained in the DO loop 19 which starts at CM49 and ends at CM179. Within this wire source loop is the observation point loop which cycles over all the appropriate observation points, wire or surface. As will be discussed later, for out of core processing the appropriate observation points are limited to a matrix block. Finally, within the observation loop there is a ground loop which sets parameters for the image of the source if present.

The general wire-wire interaction matrix element, G_{ij}, is the tangential component of electric field at the center of segment i due to a unit current at the center of segment j and zero current at the center of all other segments. Because of the sinusoidal interpolation used, the current basis function for segment j extends onto segments connected to either end of j although it is zero at the center of these segments. Rather than integrating over the entire support of the basis function for segment j in one operation to obtain the complete matrix element G_{ij} the code integrates over the

extent of segment j only, while integrating three functions simultaneously: the center of the basis function for segment j and the ends of the basic functions for adjacent segments. The values obtained represent contributions to G_{ij} and other elements G_{ik} where k is any segment connected to segment j.

In general the electric field is computed by routine INTG which assumes that the source segment is located at the origin of a cylindrical coordinate system. Thus, if segments i and j have their centers at

$$\bar{r}_i = x_i\,\hat{x} + y_i\,\hat{y} + z_i\,\hat{z}$$

$$\bar{r}_j = x_j\,\hat{x} + y_j\,\hat{y} + z_j\,\hat{z}$$

and unit vectors in the direction of the segments are

$$\hat{i} = i_x\,\hat{x} + i_y\,\hat{y} + i_z\,\hat{z}$$

$$\hat{j} = j_x\,\hat{x} + j_y\,\hat{y} + j_z\,\hat{z}$$

a cylindrical coordinate system (ρ', φ', z') is defined with origin at $\bar{r}_j$ and with $\hat{z}' = \hat{j}$. The cylindrical coordinates of segment i in this coordinate system are computed as

$$\bar{z}_{ij} = \left[(\bar{r}_i - \bar{r}_j) \cdot \hat{j} \right] \hat{j}$$

$$\bar{\rho}_{ij} = (\bar{r}_i - \bar{r}_j) - \bar{z}_{ij}$$

$$z_{ij} = \left| \bar{z}_{ij} \right| \qquad \rho_{ij} = \left| \bar{\rho}_{ij} \right|$$

The coordinates are supplied to routine INTG which returns the contributions to the matrix elements. If the segment separation distance is greater than RKH (one wavelength is default), the field is not calculated by the integration process in the routine INTG as

discussed above, but rather is calculated by the routine APRXE which uses the field expressions for a very small current element. Fields are calculated more quickly through APRXE. If a ground plane is present INTG or APRXE is also called for the image of segment j and returns the field of the image segment modified by the reflection coefficient for reflection in the ground plane. The reflection coefficients are computed in CMSET and passed to the routines that compute the field. The field of the image of segment j is added to the same matrix elements as the field of segment j. Elements are placed in the CM array by the routine MATFIL.

For the case when the observation point is a surface patch, the two tangential components of the H field of the wire are calculated by the routine HWMAT. If a perfect ground plane is present, HWMAT is called for the field of the source and image and the fields are summed. The matrix elements are placed in the CM array by MATFIL.

When the wire source loop has been exhausted, the elements corresponding to surface patches as sources are filled. This is done entirely by the two routines UDOTES and HMAT. The only function of CMSET in this case is to calculate and pass the range of rows (observation locations) in the matrix which are to be filled. When all matrix elements have been filled, a final function of the CMSET routine is to modify the diagonal wire-wire matrix elements for the case of impedance loading.

When the matrix is too large to fit into core storage, the filling operations discussed above are confined to predefined matrix blocks, and each block is written out onto file 11. The DO loop from CM34 to CM205 which encloses most of the program cycles over these matrix blocks until all the blocks are written on file 11. Examples of how a matrix might be divided into blocks are shown in figures 8 and 9; the arrows at the right of the matrix indicate points where the matrix might be divided into groups of rows. It is the transpose of the structure matrix which is actually stored in the CM array, so the blocks then become groups of columns in the transpose matrix. Due to the matrix blocking, parameters are present in the code which keep track of an element location within a block as well as the location in the total matrix.

Coding Summary:

CM22-CM30	initialization
CM34-CM205	loop over matrix blocks when matrix is stored out of core
CM49-CM179	loop over wire source segments
CM62-CM178	loop over observation points in a block
CM82-CM144	ground loop for wire observation points
CM140-CM143	calculation of wire-wire elements
CM149	placement of wire-wire elements in CM
CM152-CM160	initialization for patch observation point
CM165-CM170	ground loop for patch observation point
CM170	calculation of wire source - patch observation elements
CM172-CM176	placement of elements in CM
CM182-CM191	calculation and placement of patch source elements
CM195-CM199	modify diagonal wire elements for loading
CM201	write out matrix block for the out of core case

SYMBOL DICTIONARY:

B	= wire radius of segment j ($/\lambda$)
CAB	= array containing j_x
CABI	= i_x
CABJ	= j_x
CM	= array for storage of the transpose structure matrix
CTH	= cosine of angle between normal to ground and the reflected ray from segment j to i
DIJ	= $\hat{i} \cdot \hat{j}$
DIK	= see routine TRIO
DIL	= see routine TRIO
DIR	= $\hat{\rho}_{ij} \cdot \hat{i}$
ETA	= $\sqrt{\mu_\circ / \epsilon_\circ}$ (impedance of free space)
ETI, ETR	= arrays containing respectively the imaginary and real parts of the three contributions to the wire source matrix elements

FJ	= $\sqrt{-1}$
I	= multiple purpose index, from CM63 to CM 178 indicates observation location number in complete matrix
IFLG	= flag for routine MATFIL indicating when approximate matrix elements are used
IJ	= i-j
IK	= flag denoting surface component equation after CM154, IK = 1 for component 1 and = 0 for component 2
IM1, IM2	= first and last column number respectively of the block in the transposed matrix which is being filled
IP	= multiple purpose index, ground loop index at CM82 and CM165
IPATCH	= patch number
IPR	= loop index over columns in a transposed matrix block
ISV	= the column number of the beginning of the last block processed
IT	= number of columns in a block
IXBLK1	= do loop index for cycle over blocks
I2	= number of words in a block to be written on file 11
J	= multiple use index, CM49 to CM179 indicates source segment j
JCO1	= ICON1(J)
JCO2	= ICON2(J)
NCOL	= number of columns in matrix (number of columns in two blocks when file storage is used)
NI1	= lower do loop limit for IXBLK1 (=1 except during restart when it equals number of first block to be filled)
NROW	= number of rows in matrix (=N + 2M)
PI2	= 2 π

R = $|\bar{r}_i - \bar{r}_j|/\lambda$

REFPS = reflection coefficient for E normal to plane of incidence at CM134. Combined with REFS at CM135

REFS = reflection coefficient for E in the plane of incidence

RFL = multiplier used in constructing image segment (=1 for actual segment, = -1 for image segment)

RH = ρ_{ij}/λ

RHOSPC = distance from coordinate origin to point where reflected ray from segment i to j reflects from ground

RHOX, RHOY, RHOZ = $\bar{\rho}_{ij}/\lambda$ at CM88, and $\bar{\rho}_{ij}/|\bar{\rho}_{ij}|$ at and after CM98

RKH = separation distance when non integrated matrix elements are used (/λ)

RMAG = R

S = length of segment j (/λ)

SAB = array containing the y component of segment direction

SABI = i_y

SABJ = j_y

SALP = array containing the z component of segment direction

SALPI = i_z

SALPJ = j_z

SALPR = j_z for segment j or its image

TWHI, TWHR = arrays containing respectively the imaginary and real parts of the three contributions to the wire source matrix elements when a patch is the observation point. The first array index denotes the three components; the second denotes the surface component equation 1 or 2

T1, T2 = constants for evaluation of radial wire ground screen impedance

XIJ	$= (x_i - x_j)/\lambda$ (wire)
XJ	$= x_j/\lambda$
XSPEC	= x coordinate of ground reflection point for ray from segment i to j
XYMAG	= magnitude of projection $(\overline{r}_i - \overline{r}_j)/\lambda$ on xy plane
YIJ	$= (y_i - y_j)/\lambda$ (wire)
YJ	$= y_j/\lambda$
YSPEC	= similar to XPSEC but y component
ZIJ	$= (z_i - z_j)/\lambda$ (wire)
ZJ	$= z_j/\lambda$
ZP	$= z_{ij}/\lambda$
ZRATIS	= (impedance of ground)/(impedance of free space)
ZRSIN	= quantity used in computing reflection coefficients
ZSCRN	= quantity used in computing reflection coefficient for radial wire ground screen

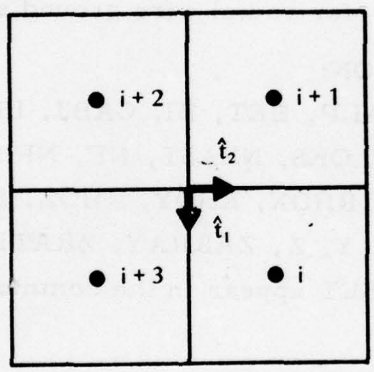
VARIABLES IN COMMON:

ALP, BET, BI, CABJ, IBLCK, ICASE, IC1, KSYMP, M, MP, N, NBLOKS, NLAST, NP, NPBLK, NRADL, PX, PY, REFPS, REFS, RHOX, RHOY, RHOZ, SABJ, SALP, SALPR, SCRWL, SCRWR, SI, X, Y, Z, ZARRAY, ZRATI, ZRATI2 (other variables not used in CMSET appear in the common blocks, check the listing).

CONECT (modified)

PURPOSE: to generate data describing the interconnection of segments by searching for segment ends that are in contact with each other or in contact with the center of a patch.

MODIFICATION: The code from CN25 to CN136 searches for connections between segments, and from segments to a ground plane. It is unchanged from that in program AMP. From CN137 to CN168 a search is made for segments connected to patches. When a connection is found the patch is divided into four patches by subroutine SUBPH (entry point to subroutine PATCH) leaving the connected segment end at the point where the four new patches meet. If the original patch is patch number i the four new patches are as shown below.



The number of each patch after the original patch number i is incremented by 3 and the total number of patches is increased by 3. In addition the connection number for the segment end (ICØN1 if end one is connected to the patch or ICØN2 if end two is connected) is set to 10000 + i. Thus a connection number greater than 10000 indicates connection to a patch and the amount by which it is greater than 10000 gives the number of the first of the four patches at the connection point.

LOCAL SYMBOL DICTIONARY: (all lengths are in units of wavelength)

I	= patch number (i)
ISEG	= segment number
IX	= location of patch data in data arrays
LD	= dimensioned length of data arrays
M	= total number of patches
SEP	= square of the separation distance between a a segment end and a patch center
SLEN	= product of the square of segment length and the square of SMIN
SMIN	= separation tolerance. A segment end is considered connected to a patch if its distance from the patch center is not greater than the product of SMIN and the segment length.
SSMIN	= square of SMIN
XI1, YI1, ZI1	= x, y and z coordinates of end one of the segment
XI2, YI2, ZI2	= x, y and z coordinates of end two of the segment
XS, YS, ZS	= x, y and z coordinates of the patch center

CONVRT (modified)

PURPOSE: to convert the segment geometry data stored as the x, y and z coordinates of each end of the segment to the x, y and z coordinates of the segment center, the segment length and two orientation angles.

MODIFICATION: The statement at CV13 has been added to cause a return to the calling program when there are patches but no segments in the model.

DATAGN (modified)

PURPOSE: Main routine for input of structure geometry data.

MODIFICATION:

DA18 to DA23:	initialization
DA31:	check whether number of patches and segments exceeds dimension limit
DA36 to DA37:	branches to set patch data
DA40:	branch to set special segment connection number
DA51 to DA59:	Subroutine PATCH is called to generate data for a surface patch.
DA60 to DA65:	PACHS (entry point to PATCH) is called to generate surface by shifting last patch input.
DA94 to DA101:	Patch dimensions are scaled by factor XW1.
DA113 to DA119:	Connection number for the segment end specified by a GC card is set to interpolate to image of the segment current.

New Mnemonics:

SP: *Defines a new surface patch*

$\left.\begin{array}{l} \text{XW1} \\ \text{YW1} \\ \text{ZW1} \end{array}\right\}$ = x, y and z coordinates of patch center

XW2 = α orientation angle of the patch normal

YW2 = β orientation angle of the patch normal

ZW2 = patch area

SS: Forms a surface by shifting the last patch input.

ITG = number of increments in x

NS = number of increments in y

XW1 = x increment

YW1 = y increment

GC: Connection number for the end of the segment specified is set equal to the segment number. This causes the segment current to be interpolated to its own image about the specified end. This card

can be used if a segment is connected to a surface other than at a patch center. It does not cause interpolation of the surface current on to the wire, however. The GC card should not be used in the normal case of a segment connected to a patch center.

ITG = segment tag number

NS = number of the segment in the set of segments having tag ITG. If ITG is zero, NS is the segment number

XW1 = segment end. If XW1 = 1. end one is affected if XW1 = 2. end two is affected.

ETMNS (modified)

PURPOSE: to fill the array representing the right hand side of the matrix equation with the negative of the electric field tangent to the segments and the tangential magnetic field on surfaces.

METHOD: The array E represents the right hand side of the matrix equation. For the i^{th} segment the right hand side is the negative of the applied electric field component tangent to the segment, and is stored in location i in array E. For the i^{th} surface patch there are two rows in the matrix equation (from the two components of the vector equations) with locations N + 2i-1 and N + 2i, where N is the total number of wire segments. The contents of E for these locations are

$$E\,(N + 2i-1) = -\hat{t}_1 \cdot (\hat{n} \times \overline{H}_i) = \pm \hat{t}_2 \cdot \overline{H}_i$$

$$E\,(N + 2i) = \hat{t}_2 \cdot (\hat{n} \times \overline{H}_i) = \pm \hat{t}_1 \cdot \overline{H}_i$$

where $\overline{H}_i$ is the magnetic field applied to patch i. The forms on the right are used in the code with the upper sign applying when $(\hat{t}_1, \hat{t}_2, \hat{n})$ forms a right hand system and the lower sign when left hand. To avoid the need to check $(\hat{t}_1, \hat{t}_2, \hat{n})$ the sign is stored in array SALP where for patch i SALP (LD + 1 - i) = $\pm$ 1 according to $(\hat{t}_1, \hat{t}_2, \hat{n})$ with LD the length of the arrays in common/DATA/.

Only minor changes have been made in the code for the electric field on segments (see reference 3). The new code for the magnetic field on patches is described below. Refer to reference 3 for symbols not defined here

ET56 to ET67: Magnetic field due to a linearly polarized plane wave is computed as

$$\overline{H}_i = \frac{\overline{H}_0}{\eta} \exp(-i\overline{k} \cdot \overline{r}_i)$$

where $\overline{H}_0 = \hat{k} \times \overline{E}_0$

LOCAL SYMBOL DICTIONARY:

ARG	$= -\overline{k} \cdot \overline{r}_i$
I	= location of patch data in data arrays
IS	= patch number
I1	= N + 2 IS-1
I2	= N + 2 IS
LD	= dimensioned length of arrays in common/DATA/
M	= total number of patches
QX, QY, QZ	= x, y and z components of the unit vector in the direction of $\overline{H}_i$
RETA	$= 1/\eta = \sqrt{\epsilon/\mu}$
SALP	= array containing sign data described above
T1X, T1Y, T1Z	= x, y and z components of $\hat{t}_1$
T2X, T2Y, T2Z	= x, y and z components of $\hat{t}_2$
T1	$= \pm \exp(-i\overline{k} \cdot \overline{r}_i)/\eta$

ET82 to ET96: Magnetic field due to an elliptically polarized plane wave is computed. The code is the same as for linear polarization except that CX, CY and CZ are used for the complex $\overline{H}_0$ and T2 is used in place of T1.

ET153 to ET164: Magnetic field due to an elementary current source at the origin and directed along the z axis of a spherical coordinate system has only a φ component given by

$$H_\varphi = \frac{I_0 \ell}{4\pi} e^{-ikR} \left(\frac{1}{R^2} + \frac{ik}{R} \right) \sin\theta$$

LOCAL SYMBOL DICTIONARY:

CX, CY, CZ	= x, y and z components of $\pm H_\varphi \hat{\varphi}$
DSH	= $I_0\, \ell/(4\pi\lambda^2)$
II	= location of patch data in data arrays
IS	= index for computing II
I1	= N + 2i - 1 where i = patch number
I2	= N + 2i
NPM	= sum of the number of segments and the number of patches
PX, PY, PZ	= x, y and z components of the unit vector in the φ direction ($\hat{\varphi}$)
P6	= $I_0\, \ell/\lambda^2$
T2	= $\pm H_\varphi$

FACTR (modified)

PURPOSE: to factor a complex matrix into a lower triangular and an upper triangular matrix using the Gauss-Doolittle technique. The factored matrix is used by subroutine SOLVE to determine the solution of the matrix equation $Ax = B$.

MODIFICATIONS: There are only small code optimization changes made in this routine. The new complex variable ARJ replaces the subscripted variable A (R, J) in the inner most do loop of steps 2 and 3 (see listing), and it replaces 1. /A(R, R) in the do loop in step 5.

FACTRS (modified)

PURPOSE: to control the factorization of the structure matrix. In particular, when structure symmetries are present, the sub-matrices are combined according to symmetric modes for factorization.

MODIFICATION: The REWIND 15 at FS119 has been added to take care of an unusual event related to the precautionary file dumps and the restarting of the program.

FFLD (modified)

PURPOSE: to calculate the far electric field, neglecting the term $\frac{e^{-ikr_0}}{r_0/\lambda}$, for wire and surface structures.

MODIFICATIONS: In the AMP code subroutine FFLD calculated the far electric field for wire structures (see reference 3); in AMP2, FFLD has been extended to include the far field due to surfaces. The actual field calculations for surface patches are performed in the subroutine FFLDS, so the main coding changes involve checking for the presence of surface patches, calling FFLDS, and summing the wire and surface fields.

A summary of the added coding follows:

FF6-FF8	modified COMMON/DATA/
FF18	addition of the complex variables EX, EY, EZ, GX, GY, GZ to be used in surface field calculations
FF27	saving of the initial value of the parameter ROZ
FF33	checking if wire segments are present, if not, wire coding is skipped
FF168	after segment fields have been calculated, the presence of surface patches is checked, if present, field calculations for surface patches are performed
FF172-FF174	initialization of pertinent variables for the case of no wire segments
FF175-FF182	initialization of variables
FF183-FF189	surface patch ground loop. Only the effects of a perfect ground are included with surface patches. (Note, for structures with wires only, the effects of real grounds are included through the use of Fresnel reflection coefficients as before)
FF186	FFLDS called for calculation of surface patch fields
FF190-FF194	summation of the wire and surface fields and calculation of the E_θ and E_φ field components

SYMBOL DICTIONARY: (new variables -- for variables not included here see reference 3)

EX, EY, EZ	= FF187 to FF189, x, y, z components of field due to surfaces (used in ground summation), FF190-FF192 segment field quantities included
GX, GY, GZ	= x, y, z components of far electric field due to surface patches returned from FFLDS
M	= number of surface patches
ROZS	= initial value of ROZ
RRZ	= $\pm$ ROZ

FFLDS

PURPOSE: to calculate the x, y, z components of the far electric field due to the surface currents; however, the term $e^{-i2\pi r_o/\lambda}/(r_o/\lambda)$ is not included.

METHOD: The expression for the far electric field for surface currents given in equation 8 can be rewritten as follows

$$\overline{E}(\overline{r}_o) = \frac{i\eta_o}{2} \frac{e^{-i2\pi r_o/\lambda}}{r_o/\lambda} \left[\hat{k}\hat{k} \cdot \int_{S_1} \overline{J}_s(\overline{r})\, e^{i2\pi\hat{k}\cdot\overline{r}/\lambda}\, dA/\lambda^2 \right.$$

$$\left. - \int_{S_1} \overline{J}_s(\overline{r})\, e^{i2\pi\hat{k}\cdot\overline{r}/\lambda}\, dA/\lambda^2 \right]$$

where $\overline{r}_o$ is the vector from the origin of coordinates to the observation point, and $\hat{k}$ is in the $\overline{r}_o$ direction. Note that the integrals in the above expression are identical. The integral is calculated as a simple sum over the surface patches. The quantity which is returned to the calling program is $\frac{r_o/\lambda}{e^{-i2\pi r_o/\lambda}} \overline{E}(\overline{r}_o)$.

SYMBOL DICTIONARY:

ARG	$= 2\pi\hat{k}\cdot\overline{r}/\lambda$
CONS	$= i\eta_o/2$
CT	= multiple use complex variable, $e^{i2\pi\hat{k}\cdot\overline{r}/\lambda}\Delta A/\lambda^2$ at FL22, $\hat{k}$ dot the integral at FL28
EX, EY, EZ	= x, y, z components of integral summation at FL24, equal to $\frac{r_o/\lambda}{e^{-i2\pi r_o/\lambda}} \overline{E}(\overline{r}_o)$ at FL30
I	= array location of patch data
J	= loop index over patches
K	= current array index

ROX, ROY, ROZ	= x, y, z components of $\hat{k}$
S	= $\Delta A/\lambda^2$ (the area of a patch)
SCUR	= array where the x, y, z components of surface current are stored
TPI	= 2π
XS, YS, ZS	= arrays containing center point coordinate of patches in wavelengths

VARIABLES IN COMMON:

BI, LD, M, X, Y, Z (other variables in common not referenced in FFLDS, see listing).

GH

PURPOSE: to compute the function which is numerically integrated for the near H field of a segment.

METHOD: The value returned by GH is

$$G = \left[\frac{1}{(kr)^3} + \frac{i}{(kr)^2} \right] e^{-ikr}$$

$$\text{where } r = \left[\rho'^2 + (z - z')^2 \right]^{1/2}$$

ρ' = ρ coordinate of the field observation point in a cylindrical coordinate system with origin at the center of the source segment and z axis oriented along the source segment.

z' = z coordinate of the field observation point in the cylindrical coordinate system.

z = z coordinate of the integration point on the source segment.

k = $2\pi/\lambda$

SYMBOL DICTIONARY:

CKR	= cos (kr)
HR	= real part of G
HI	= imaginary part of G
R	= kr
RHKS	= $(k\rho')^2$
RR2	= $1./(kr)^2$
RR3	= $1./(kr)^3$
RS	= $(kr)^2$
SKR	= sin (kr)
ZK	= kz
ZPK	= kz'

HFK

PURPOSE: to compute the near H field of a uniform current filament by numerical integration.

METHOD: The H field of a current filament of length Δ with uniform current distribution of magnitude $I = \lambda$ is

$$H_\varphi = \frac{k\rho'}{2} \int_{-k\frac{\Delta}{2}}^{k\frac{\Delta}{2}} \left[\frac{1}{(kr)^3} + \frac{i}{(kr)^2} \right] e^{-ikr} d(kz)$$

where r, ρ' and z are defined in the description of subroutine GH. The numerical integration is performed by the method of Romberg quadrature with variable interval width, which is described in the discussion of subroutine INTX in reference 3. The integral is multiplied by $k\rho'/2$ at HK82 and HK83 in the code.

SYMBOL DICTIONARY:

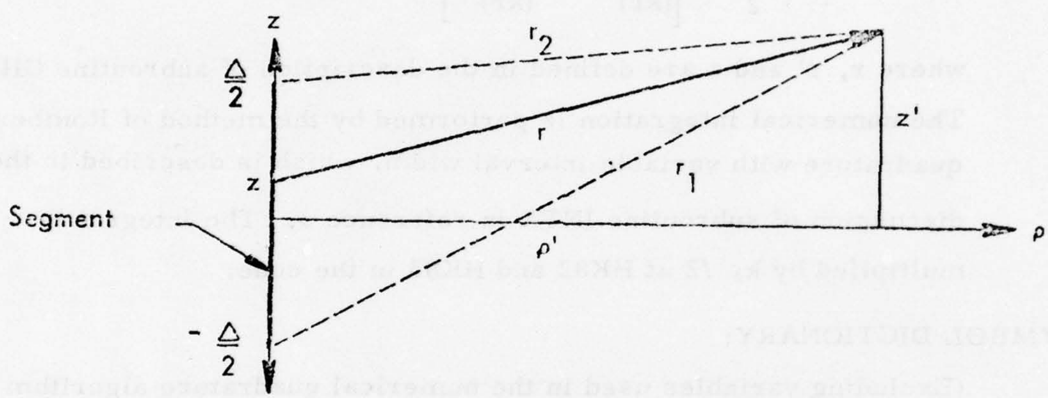
(Excluding variables used in the numerical quadrature algorithm which are defined under subroutine INTX in reference 3.)

RHK	$= k\rho'$
RHKS	$= (k\rho')^2$
SGI	= imaginary part of H_φ
SGR	= real part of H_φ
ZPK	$= kz'$ (z' = z coordinate of observation point)
ZPKX	= ZPK

HFLD

PURPOSE: to compute the near H field of filamentary currents of sine, cosine and constant distribution on a segment.

METHOD: The wire segment is considered to be located at the origin of a local cylindrical coordinate system with the point at which the field is computed being (ρ', φ', z'). The geometry for a filament of current of length Δ is shown below



For a sine or cosine current distribution the field can be written in closed form. For a current $I_0 \begin{pmatrix} \sin kz \\ \cos kz \end{pmatrix}$ the field is

$$H_\varphi(\rho', z') = \frac{-iI_0/\lambda}{2k\rho'}\left[e^{-ikr_2}\begin{pmatrix}\cos k\Delta/2 \\ -\sin k\Delta/2\end{pmatrix} - e^{-ikr_1}\begin{pmatrix}\cos k\Delta/2 \\ \sin k\Delta/2\end{pmatrix}\right.$$

$$-i\left(kz' - k\frac{\Delta}{2}\right)\frac{e^{-ikr_2}}{kr_2}\begin{pmatrix}\sin k\frac{\Delta}{2} \\ \cos k\frac{\Delta}{2}\end{pmatrix}$$

$$\left. + i\left(kz' + k\frac{\Delta}{2}\right)\frac{e^{-ikr_1}}{kr_1}\begin{pmatrix}-\sin k\frac{\Delta}{2} \\ \cos k\frac{\Delta}{2}\end{pmatrix}\right]$$

$I_0/\lambda = 1$ is assumed in this routine.

The field due to a constant current is obtained by numerical integration which is performed by subroutine HFK. If ρ' is zero all field quantities are set to zero since H_φ is undefined.

SYMBOL DICTIONARY:

CDK	$= \cos(k\Delta/2)$
CONS	$= -i/(2k\rho')$
DH	$= \Delta/(2\lambda)$
DK	$= k\,\Delta/2$
EKR1	$= e^{-ikr_1}$
EKR2	$= e^{-ikr_2}$
FJ	$= i = \sqrt{-1}$
FJK	$= -i\,2\pi$
HKI	= imaginary part of HPK
HKR	= real part of HPK
HPC	= H_φ due to a cosine current distribution
HPK	= H_φ due to a constant current distribution
HPS	= H_φ due to a sine current distribution
RH	$= \rho'/\lambda$
RH2	$= (\rho'/\lambda)^2$
R1	$= r_1/\lambda$
R2	$= r_2/\lambda$
S	$= \Delta/\lambda$
SDK	$= \sin(k\Delta/2)$
TP	$= 2\pi$
T1	$= (kz' + k\frac{\Delta}{2})e^{-ikr_1}/kr_1$
T2	$= (kz' - k\frac{\Delta}{2})e^{-ikr_2}/kr_2$
ZP	$= z'/\lambda$
Z1	$= (z' + \Delta/2)/\lambda$
Z2	$= (z' - \Delta/2)/\lambda$

HINTG

PURPOSE: to calculate the H field due to one patch observed at another, and to calculate the resulting interaction matrix element.

METHOD: The $\overline{H}$ field due to patch j at patch i is calculated by the expression

$$\overline{H}_{ij} = \frac{-1}{4\pi}\left[(1 + i\,2\pi\frac{R}{\lambda}{}_{ij})\ \frac{e^{-i2\pi\frac{R}{\lambda}ij}}{\left(\frac{R_{ij}}{\lambda}\right)^3}\right]\left(\frac{\overline{R}}{\lambda}{}_{ij}\right) \times \overline{J}_j\ \frac{\Delta A}{\lambda^2}{}_j \qquad (10)$$

where $\overline{R}_{ij} = \overline{r}_i - \overline{r}_j$ and ΔA_j is the area of patch j. This expression is equal to the kernel of the integral in equation (2) evaluated at the appropriate patch center point multiplied by the patch area; in addition the factor $\frac{1}{4\pi}$ is included.

The components of equation (10) in the $-\hat{t}_{1i}$ and $-\hat{t}_{2i}$ directions are calculated for elements of the interaction matrix. For the case of a perfectly conducting ground plane, the subroutine HINTG is called twice to calculate the patch and patch image field, and the appropriate field components are summed for this case in HINTG.

SYMBOL DICTIONARY:

CR	$= \cos(-kR)$
FPI	$= 4\pi$
F1X, F1Y, F1Z	$= -\overline{H}_{ij}$ with J_{1j} as the current source. Note $\overline{J}_j = J_{1j}\,\hat{t}_{1j} + J_{2j}\,\hat{t}_{2j}$
F2X, F2Y, F2Z	$= -\overline{H}_{ij}$ with J_{2j} as the current source
GAM	$= \frac{1}{4\pi}(1 + ikR)\,\frac{e^{-ikR}}{\left(\frac{R}{\lambda}\right)^3}\,\frac{\Delta A}{\lambda^2}$

GX, GY, GZ	$= GAM * \overline{R}/\lambda$
G11	$= -\hat{t}_{1i} \cdot \overline{H}_{ij1}$ where $\overline{H}_{ij1}$ is the field with J_{1j} as the source
G12	$= -\hat{t}_{1i} \cdot \overline{H}_{ij2}$
G21	$= -\hat{t}_{2i} \cdot \overline{H}_{ij1}$
G22	$= -\hat{t}_{2i} \cdot \overline{H}_{ij2}$
II	= observation patch number
IP	= image flag: 1 for structure patch, 2 for image patch
J	= source patch location in array
JJ	= source patch number
R	$= R/\lambda$
RFL	= parameter used in image calculation: 1 for structure patch, -1 for image patch
RK	$= -2\pi R/\lambda$
RSQ	$= (R/\lambda)^2$
RX, RY, RZ	$= \overline{R}/\lambda$
S	$= \Delta A/\lambda^2$
SR	$= \sin(-kR)$
TPI	$= 2\pi$
T1QX, T1QY, T1QZ	$= \hat{t}_1$ for a patch
T1X, T1Y, T1Z	= arrays containing the components of $\hat{t}_1$
T2QX, T2QY, T2QZ	$= \hat{t}_2$ for a patch

$$\left.\begin{array}{l}\text{T2X}\\ \text{T2Y}\\ \text{T2Z}\end{array}\right\} = \text{arrays containing the components of } \hat{t}_2$$

VARIABLES IN COMMON:

ALP, BET, BI, ICON1, ICON2, IPSYM, ITAG, LD, M, MP, N, NP, SALP, SI, T1XI, T1YI, T1ZI, T2XI, T2ZI, WLAM, X, XI, Y, YI, Z, ZI.

HMAT

PURPOSE: to fill the matrix elements representing the interaction between surface patches.

METHOD: Subroutine HMAT has as input the column range (IM1 to IM2) in the transposed structure matrix between which the surface to surface interaction elements are to be filled. For the most part the matrix elements themselves are calculated in subroutine HINTG; thus, HMAT calls HINTG to obtain the elements for each source-observation patch pair and places them in the appropriate locations in the transposed structure matrix, CM. The self terms on the other hand appear simply as $\pm 1/2$ and are added directly into the matrix by HMAT. Each observation patch accounts for two columns in CM as indicated by the two component equations in equation (4); therefore, in dividing the matrix into blocks for out of core processing these columns may appear in different blocks, and extra coding is present to check and take care of this situation.

The sign of certain matrix elements is dependent on the symmetry used in structure construction. This happens since the t vectors are reflected through a plane of symmetry and the conditions $\hat{t}_1 \times \hat{n} = -\hat{t}_2$ and $\hat{t}_2 \times \hat{n} = \hat{t}_1$ on the first side of the symmetry plane become $\hat{t}_1 \times \hat{n} = \hat{t}_2$ and $\hat{t}_2 \times \hat{n} = -\hat{t}_1$ on the other side. The result is sign changes in equation 4a and 4b; these changes are made in the code through the SALP variable which is 1 for the former case and -1 for the latter.

The subroutine coding is as follows: parameter set up HM19 to HM28, observation loop HM30 to HM82, source loop HM46 to HM82 divided into an outer symmetry period loop and an inner loop over patches in period, ground loop HM 56 to HM57, and matrix element placement HM58 to HM81.

SYMBOL DICTIONARY:

CM	= transpose structure matrix array
G11 G12 G21 G22	= matrix elements from HINTG
IL	= observation patch array location
IM1	= column number where fill begins
IM2	= column number where fill ends
IP	= patch number loop index
IPEND	= last patch number within column range
IPST	= first patch number within column range
ISELEN	= flag denoting whether component equation 1 or 2 is the last column in the matrix block
ISELST	= flag denoting whether component equation 1 or 2 is the first column in the matrix block
ISTART	= first matrix column to be filled by HMAT
IX	= ground loop index
I1	= column number in block for component 1 equation
I2	= column number in block for component 2 equation
J	= source patch number
J1 J2	= row numbers for elements corresponding to the source current components 1 and 2 respectively
K	= matrix block column number index
L	= symmetry period loop index
LL	= index of loop over patches in a period of symmetry
NCOL	= number columns in transposed structure matrix (equal to the total number of equations for one symmetric section)
NOP	= number of periods of symmetry
NROW	= number of rows in matrix (equal to total number of unknowns)
T1X T1Y T1Z	= $\hat{t}_1$ array

$\left.\begin{array}{l}\text{T2X}\\ \text{T2Y}\\ \text{T2Z}\end{array}\right\}$ = $\hat{t}_2$ array

$\left.\begin{array}{l}\text{XS}\\ \text{YS}\\ \text{ZS}\end{array}\right\}$ = *array for the coordinates of patch centers in wavelengths*

CONSTANTS: $\pm$ 1/2 = matrix self terms

VARIABLES IN COMMON:

ALP, BET, BI, CH, CL, ICON1, ICON2, IFAR, IPERF, IPSYM, ITAG, KSYMP, LD, M, MP, N, NP, NRADL, SALP, SCRWL, SCRWR, SI, T1XI, T1YI, T1ZI, T2XI, T2YI, T2ZI, WLAM, X, XI, Y, YI, Z, ZI, ZRATI, ZRATI2.

HWMAT

PURPOSE: to compute the matrix elements associated with the H field at a surface patch due to the current on a wire segment.

METHOD: Subroutine HWMAT is called by subroutine CMSET to compute the H field tangent to patch i produced by three components of the current basis functions on segment j. The three integrated functions on segment j produce six matrix element contributions for the case of the surface patch since the field of each current function is decomposed along the two tangent vectors. The expressions for the matrix element contributions are identical to those presented in the INTG discussion in reference 3 if H is substituted for E, and in addition, the dot products are taken along $\mp\hat{t}_2$ and $\mp\hat{t}_1$, the surface tangent vectors, rather than a wire direction vector. (INTG calculates the matrix elements representing the E field at a segment i due to the current on a segment j). The upper sign on the surface tangent vectors applies for the patch where $(\hat{t}_1, \hat{t}_2, \hat{n})$ form a right hand system (normal case) and the lower sign applies for a left hand system (patch reflected in a symmetry plane). This can be seen in the derivation of the scalar components in equation 4, and the matrix elements being discussed here arise from the wire terms in equation 4. In addition equation 4b has been multiplied through by a minus sign for usage in the code.

The H field of the segment j for a sine, cosine, and constant current is calculated by the subroutine HFLD. The field is calculated for the segment located at the origin of a cylindrical coordinate system, so the cylindrical ρ and z coordinates of the observation point are passed to HFLD and the phi components of the field for sine, cosine, and constant current are returned to HWMAT. (The phi component is the only non zero component). The matrix elements are then calculated with the expressions discussed above. When the source and observation points are separated by a distance greater than a specified value (RKH), the field terms

are no longer calculated by HFLD. Rather the routine APRXH is called which calculates the field without integration and returns the proper matrix elements to HWMAT.

For the case of a perfectly conducting ground plane, HWMAT is called to calculate the matrix element contribution for the source segment image as well as the source segment. The contribution of the image is added to the same matrix element as the actual source segment.

A summary of the coding follows:

HW22-HW29	calculation of cylindrical coordinates of the observation point
HW32-HW37	calculation of the x, y, z components of $\hat{\varphi}$ in the cylindrical system centered on the source segment
HW41	calculation of H field
HW53-HW72	loop over the two equation components
HW54-HW59	H field components
HW60-HW71	calculation of matrix elements

SYMBOL DICTIONARY:

CABJ	= x component of source segment direction
CK, CL, CONS, COSK, COSL, DIK, DIL	= see INTG

HCDT	$= \mp \hat{t}_{2i} \cdot \overline{H}_c$ or $\mp \hat{t}_{1i} \cdot \overline{H}_c$ where H_c is the field due to a cosine current
HKDT	$= \mp \hat{t}_{2i} \cdot \overline{H}_k$ or $\mp \hat{t}_{1i} \cdot \overline{H}_k$ where H_k is the field due to a constant current
HPC HPK HPS	= field due to the cosine, constant, and sine current components respectively in the cylindrical coordinate system
HSDT	$= \mp \hat{t}_{2i} \cdot \overline{H}_s$ or $\mp \hat{t}_{1i} \cdot \overline{H}_s$ where H_s is the field due to a sine current
IFLG	= flag returned to CMSET denoting use of HFLD or APRXH
ILC	= array location of patch data
IP	= ground flag
IPATCH	= patch number
K	= index for component equations
PDT	$= \pm \hat{t}_2 \cdot \hat{\varphi}$ or $\pm \hat{t}_1 \cdot \hat{\varphi}$, the sign is determined by contents of SALP array. For a right handed system SALP = +, and left handed SALP = -.
PX PY PZ	= x, y, z components of the cylindrical $\hat{\varphi}$ (cylindrical coordinate system centered at source segment)
R	= source-observation separation distance ($/\lambda$)
RFL	= multiplier creating image segment
RH	$= \rho$
RHX RHY RHZ	$= \hat{\rho}$ after HW31
R2	$= R^2$
SABJ	= y component of source segment direction
SALP	= see PDT

SALPJ = z component of source segment or image direction

SALPT = z component of source segment direction

SILK, SINK, SINL = see INTG

SJ = source segment length

TP = 2π

TWHI, TWHR = arrays containing the imaginary and real parts respectively of the matrix elements. The first index of the array refers to the current basis functions, the second index refers to the scalar component equations 4a and 4b.

T1X, T1Y, T1Z = arrays containing the x, y, z components of $\hat{t}_1$

T2X, T2Y, T2Z = arrays containing the x, y, z components of $\hat{t}_2$

XD = $(\bar{r}_o - \bar{r})_x/\lambda$

XJ = r_x/λ, the x component of the source segment

XS = array containing x coordinates of patch centers $(/\lambda)$

YD = $(\bar{r}_o - \bar{r})_y/\lambda$

YJ = r_y/λ

YS = array containing y coordinates of patch centers $(/\lambda)$

ZD = $(\bar{r}_o - \bar{r})_z/\lambda$

ZJ = r_z/λ of source and image

ZP = z in the cylindrical coordinate system

ZS = array containing z coordinates of patch centers $(/\lambda)$

ZT = r_z/λ of source (input)

VARIABLES IN COMMON:

ALP, BET, ICON1, ICON2, SALP, SI, X, Y, Z

(other variables appear in common not referenced, see listing).

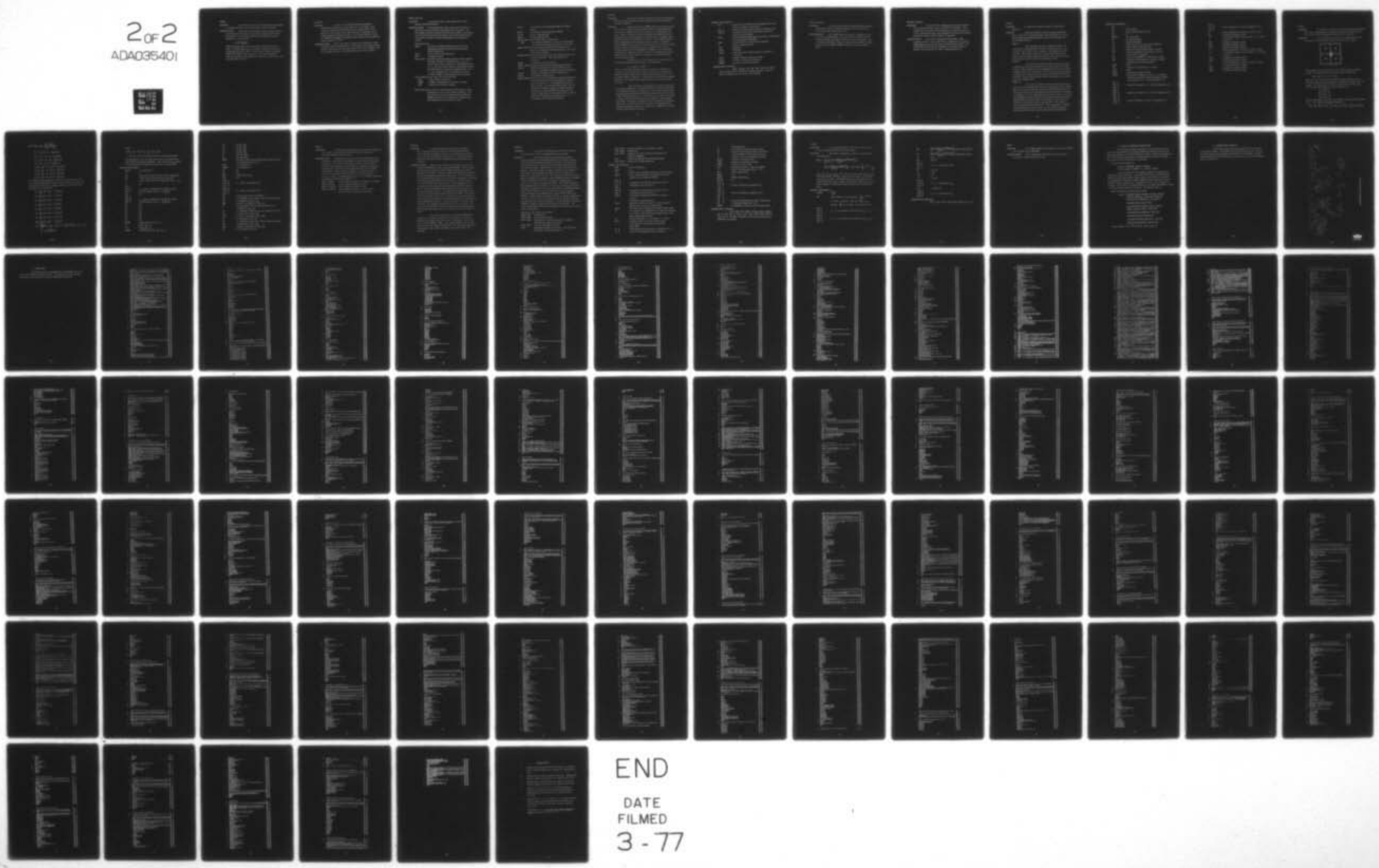
ISEGNO (modified)

PURPOSE: to determine the segment number of the m^{th} segment ordered by increasing segment numbers in the set of segments with tag numbers equal to the given tag number.

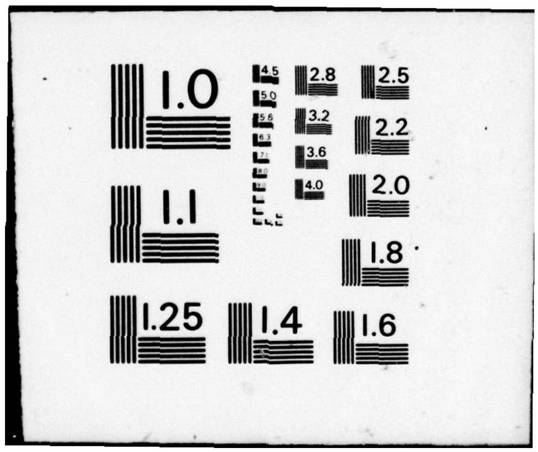
MODIFICATIONS: The formal parameter M in AMP has been changed to MX in AMP2 to avoid conflict with the variable name M in the new common/DATA/array. The MX variable occurs at IS1, IS9, IS14, and IS20. In addition, execution is terminated for the case of no wire segments since this would be an invalid call. The segment condition is checked at IS16.

AD-A035 401 MB ASSOCIATES SAN RAMON CALIF F/G 9/5
ANTENNA MODELING PROGRAM SUPPLEMENTARY COMPUTER PROGRAM MANUAL --ETC(U)
JAN 75 N00014-71-C-0187
UNCLASSIFIED MB-R-75/4 NL

2 OF 2
ADA035401



END
DATE
FILMED
3-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

JMELS

PURPOSE: to add matrix element contributions due to segments at multiple junctions into the appropriate matrix locations.

MODIFICATION: The form of the structure matrix for the case of a wire and surface structure can be seen in figures 8 and 9; thus, the location, ℓ, of a matrix element associated with a source segment, j, can be written

$$\ell = 2\,MP \left[\frac{j-1}{NP}\right] + j$$

where the integer part of the expression inside the bracket is taken. This calculation has been added to JMELS in the form of a statement function at JM13, and the matrix indices calculated at JM16 and JM21 use this function. Previously the matrix indices were equal to the appropriate segment numbers; this is the case when MP = 0 above.

LFACTR

PURPOSE: to perform the Gauss-Doolittle factorization calculations on two blocks of the matrix in core storage. This routine in conjunction with FACIO factors a matrix which is too large for core storage into an upper and lower triangular matrix using the Gauss-Doolittle technique. The factored matrix is used by LUNSCR and LTSOLV to determine the solution of the transposed matrix equation $x^T A^T = B^T$.

MODIFICATIONS: Minor optimization changes were made in the code. The new complex variable AJR is used in place of the array element A(J, R) in the innermost DO loop of steps 2 and 3 (see listing). AJR is used again in place of 1. /A(J2P1, R) in the DO loop of step 5.

MAIN (modified)

PURPOSE: to control the input, output and the flow of the Antenna Modeling Program.

MODIFICATIONS: The modifications to the main code are simple in nature and will be discussed individually in the list below. All the changes relate to the three new functions of AMP2: calculations involving wire and surface structures, use of time saving approximate matrix elements, and the precautionary file dumping for restart.

Modification list:

MA33	inclusion of the KH and DP mnemonics for the approximate matrix element and file dump data cards respectively
MA43	call for time at execution start
MA71	no segment check
MA87-MA97	printing of geometry data related to surface patches. The patch normal is computed from $\hat{n} = \pm \hat{t}_1 \times \hat{t}_2$ where the plus sign is used for a right handed system (t_1, t_2, n), and a minus sign for a left handed system; this information is stored in the SALP array. The x, y, z components of $\hat{n}$ are temporarily stored in TMP1, TMP2, TMP3 respectively for printing
MA102-MA108	minor data checking and defining
NEQ	= total number of unknowns
NPEQ	= number of unknowns in a symmetric section
NOP	= number of symmetric sections
MA109-MA133	this section of code determines the matrix blocking parameters for out of core matrix problems. The size of the matrix is now determined by NEQ and NPEQ and these two variables have replaced N and NP respectively in this section of code

MA160	initialization of the parameter RKH (see MA213 below)
MA172	initialization of the parameter IDUMP
MA193	test for mnemonic KH
MA195	test for mnemonic DP
MA213-MA219	setting RKH equal to input value
RKH	= the separation distance criterion in wavelengths when approximate matrix elements will be used.
MA255-MA257	The only ground allowable when using surface patches is a perfectly conducting ground. This code checks that condition
MA258-MA360	near field calculations are not allowed for the case of surface patches. This code checks this condition
MA454	no segment condition checked
MA461-MA468	frequency scaling of patch geometry parameters
MA506	NROW substituted for N in the CMSET calling sequence, and RKH is added to the calling sequence
MA513	NPEQ is substituted for NP in FACTRS call
MA593	no segment condition check
MA624-MA641	This section of code prints the magnitude and phase of the two surface current components on patches, and it prints the x, y, z components of the total patch current. The variables ETH, EPH, EX, EY, and EZ are used as temporary storage during the calculations. ETH and EPH are set equal to the current components 1 and 2 respectively on a patch while EX, EY and EZ are used for the x, y, z components of the total patch current.

MATFIL

PURPOSE: to fill the CM array with matrix elements representing fields due to wire segment currents (either E field on a segment or H field on a patch).

METHOD: As discussed in the CMSET writeup, the basis function for the source segment current extends onto the adjacent segments, but the entire support of the basis function is not integrated at one time. Rather, three functions are integrated simultaneously when integrating over a given segment; one function for the segment itself, and the other two for the adjacent segments. Thus, MATFIL places these contributions into the proper matrix elements or calls the routine JMELS for the case of multiple segments connected to an end. When the matrix contribution has been computed by the current element expression (routine APRXE or APRXH), only one element contribution is obtained, and this element corresponds to the source segment itself. This condition is signaled to the MATFIL routine by the parameter IFLG.

The proper row in the transpose structure matrix for contributions related to the segment j is computed by

$$\ell = 2\ \mathrm{MP} \left[j - 1/\mathrm{NP} \right] + j$$

where ℓ is the matrix location, MP is the number of patches in a symmetric section, NP is the number of segments in a symmetric section, and the segments are numbered consecutively from 1 to N the total number of segments. The brackets imply taking the integer part here. The origin of the expression can be seen by referring to figure 9.

The sign of the contributions to the matrix elements of segments adjacent to j is determined by the reference directions of the adjacent segments compared to j. The connection data of the adjacent segments (stored in the ICON arrays) is checked in order to determine the proper sign. For the case of a segment connected a surface, the connection parameter is assigned a value greater than 10000. This case is treated in the same manner as a segment connected to a perfect ground; the current is interpolated to the segment image which means the contribution is added to the matrix element associated with segment j.

SYMBOL DICTIONARY:

CM	= array for core storage of the transposed structure matrix, or blocks of the matrix
ETI ETR	= arrays containing respectively the imaginary and real parts of the three contributions to the wire source matrix elements
IFLG	= flag indicating whether one or three term contributions are being passed to MATFIL
IPR	= column number in CM array for elements
J	= source segment number
JCO1	= ICON1(J)
JCO2	= ICON2(J)
JL(K)	= statement function determining row position in CM array
JLOC	= integer variable used as row index
NCOL	= number of columns in CM array
NROW	= number of rows in CM array

VARIABLES IN COMMON:

ICON1, ICON2, JIX, JIZ, JOX, JOZ, MP, NCIX, NCIZ, NCOX, NCOZ, NP (other variables appear in common blocks in MAFIL, but are not used, see listing).

MOVE (modified)

PURPOSE: to rotate and translate previously defined segments and patches, either moving the original structure or leaving it fixed and producing duplicates of it.

MODIFICATION: The transformation of segment coordinates is the same as in program AMP. The new code from MO64 to MO103 transforms patch coordinates in the same way as segments. The coordinates of the patch center are transformed at MO80 to MO82, the vectors $\hat{t}_1$ at MO86 to MO88, and the vectors $\hat{t}_2$ at MO92 to MO94.

NETWK (modified)

PURPOSE: To solve for the voltages and currents at the ports of non-radiating networks which are part of the antenna. This routine also is involved in the solution for current when there are no non-radiating networks, and computes the relative driving point matrix asymmetry when this option is requested.

MODIFICATION: Only minor changes have been made. NEQ, computed at NT19, is the total number of equations to be solved. NEQ replaces N as a do loop limit at several places and all calls to SOLVES have the number of patches, M, and the number of patches in a symmetric cell, MP, in the parameter list.

PATCH

PURPOSE: to define and modify the data for surface patch geometry.

METHOD: Subroutine PATCH consists of three independent parts. The first is entered through a call to PATCH, the second through entry point PACHS at PA45 and the third through entry point SUBPH at PA90.

PART 1:

The first section defines a single new patch with center point coordinates (XC, YC, ZC), normal vector $\hat{n} = \cos\alpha \cos\beta\,\hat{x} + \cos\alpha\,\sin\beta\,\hat{y} + \sin\alpha\,\hat{z}$ where α = AL and β = BT, and area AR. Patch data is stored in common block/DATA/in the same arrays as segment data but starting at the tops of the arrays and filling decreasing locations as patches are input.

PART 2:

The section from PA45 to PA86 forms a flat surface of patches by shifting the last patch previously entered in the data arrays. NX new patches are first generated with successive shifts of distance XC in the x direction. Then NY new rows are generated by shifting the resulting row of patches by the distance YC in the y direction. Areas and orientations of the new patches are the same as for the original patch.

PART 3:

The code from PA90 to PA147 divides patch number NX into four new patches each having 1/4 of the area of the original patch. The four new patches are numbered NX through NX+3 and all patches following the original patch number NX are incremented in number by 3. Since patch data is stored in the arrays of common block/DATA/from the top down with increasing number the data for patches NX+1 through M must be shifted down in the arrays by 3 locations to leave room for the 3 new patches. This is done in the code from PA94 through PA109. The original patch is divided along lines parallel to the vectors $\hat{t}_1$ and $\hat{t}_2$ and the new patch data stored in data array in statements PA110 through PA147.

SYMBOL DICTIONARY:

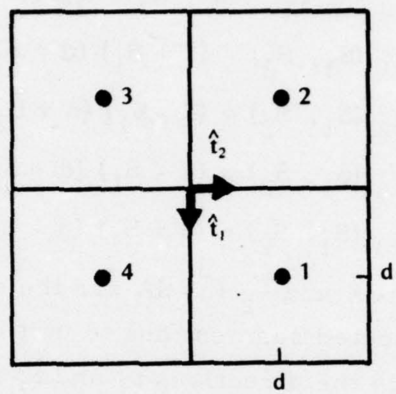
AL	= α
AR	= area of patch
BI	= array containing patch areas
BT	= β
IPSYM	= symmetry flag
IX	= do loop parameter
IY	= do loop parameter
LD	= length of arrays in common block/DATA/
M	= total number of patches input
MI	= index for location of patch data in arrays
MP	= number of patches in a symmetric section
N	= total number of segments
NP	= number of segments in a symmetric section
NX	= number of increments in x in part 2, number of the patch to be divided in part 3
NXP	= NX+1
NY	= number of increments in Y in part 2
NYP	= NY+1
SALN	= temporary storage variable
SALP	= data array for patches. If ($\hat{t}_1$, $\hat{t}_2$, $\hat{n}$) forms a right hand coordinate system the value in SALP for that patch is +1., if left hand SALP is -1.
S1X, S1Y, S1Z	= temporary storage for x, y and z components of $\hat{t}_1$.
S2X, S2Y, S2Z	= temporary storage for x, y and z components of $\hat{t}_2$.
T1X, T1Y, T1Z	= arrays containing x, y and z components of $\hat{t}_1$

T2X, T2Y, T2Z	= arrays containing x, y and z components of $\hat{t}_2$
X	= array containing x coordinates of patch centers
XA	= temporary storage variable
XC	= x coordinate of patch center in part 1, x increment in part 2
XST	= temporary storage variable
XW1	= temporary storage variable
XW2	= temporary storage variable
Y	= array containing y coordinates of patch centers
YC	= y coordinate of patch center in part 1, y increment in part 2
YW1	= temporary storage variable
YW2	= temporary storage variable
Z	= array containing z coordinates of patch centers
ZC	= z coordinate of patch center
ZW1	= temporary storage variable
ZW2	= temporary storage variable

PCINT

PURPOSE: to compute the interaction matrix elements representing the electric field tangent to a segment connected to a surface due to the current on the four patches around the connection point.

METHOD: The four patches at the base of a connected wire are located as shown below with respect to the vectors $\hat{t}_1$ and $\hat{t}_2$



where patch numbers indicate the order of the patches in the data arrays. The position of a point on the surface is defined by

$$\bar{\rho}\,(S_1,\ S_2) = \bar{\rho}_0 + S_1\,\hat{t}_1 + S_2\,\hat{t}_2$$

where $\bar{\rho}_0$ is the position of the center of the four patches, where the wire connects, and S_1 and S_2 are coordinates measured from the center. Representing the current over the surface by $\bar{J}\,(S_1,\ S_2)$ the currents at the centers of the four patches are

$$\bar{J}_1 = \bar{J}\,(d,\ d)$$
$$\bar{J}_2 = \bar{J}\,(-d,\ d)$$
$$\bar{J}_3 = \bar{J}\,(-d,\ -d)$$
$$\bar{J}_4 = \bar{J}\,(d,\ -d)$$

and the current at the center of the segment, flowing onto the surface is I_0. The current interpolation function is

$$\bar{J}\,(S_1,\ S_2) = \left[\bar{f}\,(S_1,\ S_2) - \sum_{i=1}^{4} g_i\,(S_1,\ S_2)\ \bar{f}_i\right] I_0 + \sum_{i=1}^{4} g_i\,(S_1,\ S_2)\,\bar{J}_i$$

where $\overline{f}(S_1, S_2) = \dfrac{S_1\hat{t}_1 + S_2\hat{t}_2}{2\pi(S_1^2 + S_2^2)}$

$$\overline{f}_1 = \overline{f}(d, d) = (\hat{t}_1 + \hat{t}_2)/(4\pi d)$$

$$\overline{f}_2 = \overline{f}(-d, d) = (-\hat{t}_1 + \hat{t}_2)/(4\pi d)$$

$$\overline{f}_3 = \overline{f}(-d, -d) = (-\hat{t}_1 - \hat{t}_2)/(4\pi d)$$

$$\overline{f}_4 = \overline{f}(d, -d) = (\hat{t}_1 - \hat{t}_2)/(4\pi d)$$

$$g_1(S_1, S_2) = (d + S_1)(d + S_2)/(4d^2)$$

$$g_2(S_1, S_2) = (d - S_1)(d + S_2)/(4d^2)$$

$$g_3(S_1, S_2) = (d - S_1)(d - S_2)/(4d^2)$$

$$g_4(S_1, S_2) = (d + S_1)(d - S_2)/(4d^2)$$

If $\overline{\Gamma}_1(\overline{\rho})\,dA$ and $\overline{\Gamma}_2(\overline{\rho})\,dA$ are the electric fields at the center of the connected segment due to unit currents at $\overline{\rho}$ on the surface dA, flowing in the directions $\hat{t}_1$ and $\hat{t}_2$ respectively, the nine matrix elements to be computed are

$$E_1 = \int_S g_1(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_1(\overline{\rho})\,dA$$

$$E_2 = \int_S g_2(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_1(\overline{\rho})\,dA$$

$$E_3 = \int_S g_3(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_1(\overline{\rho})\,dA$$

$$E_4 = \int_S g_4(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_1(\overline{\rho})\,dA$$

$$E_5 = \int_S g_1(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_2(\overline{\rho})\,dA$$

$$E_6 = \int_S g_2(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_2(\overline{\rho})\,dA$$

$$E_7 = \int_S g_3(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_2(\overline{\rho})\,dA$$

$$E_8 = \int_S g_4(S_1, S_2)\,\hat{i}\cdot\overline{\Gamma}_2(\overline{\rho})\,dA$$

$$E_9 = \int_S \left\{\left[\left(\overline{h}(S_1, S_2)\cdot\hat{t}_1\right]\left[\hat{i}\cdot\overline{\Gamma}_1(\overline{\rho})\right] + \left[\overline{h}(S_1, S_2)\cdot\hat{t}_2\right]\right.\right.$$
$$\left.\left.\left[\hat{i}\cdot\overline{\Gamma}_2(\overline{\rho})\right)\right]\right\}\,dA$$

where

$$\bar{h}(S_1, S_2) = \bar{f}(S_1, S_2) - \sum_{i=1}^{4} g_i(S_1, S_2)\,\bar{f}_i$$

$\hat{i}$ = the unit vector in the direction of the connected segment.

The integration is over the total area of the four patches and is performed by numerical quadrature. The number of increments in S_1 and S_2 used in integration is set by the variable NINT.

SYMBOL DICTIONARY:

CABI	= x component of $\hat{i}$
D	= d
DA	= area of the surface element used in integration
DS	= width of the surface element of area DA
E	= array used to return the values E_1, E_2, ... E_9
E1	= E_1
E1X, E1Y, E1Z	= x, y and z components of $\bar{\Gamma}_1(\bar{\rho})$ DA at PC46. At PC47 E1X is set to $\hat{i} \cdot \bar{\Gamma}_1(\bar{\rho})$ DA
E2	= E_2
E2X, E2Y, E2Z	= x, y and z components of $\bar{\Gamma}_2(\bar{\rho})$ DA at PC46. At PC48 E2X is set to $\hat{i} \cdot \bar{\Gamma}_2(\bar{\rho})$ DA.
E3	= E_3
E4	= E_4
E5	= E_5
E6	= E_6
E7	= E_7
E8	= E_8
E9	= E_9
FCØN	= $1/(4\pi d)$ factor in $\bar{f}_1$, $\bar{f}_2$ ...
F1	= $\bar{h}(S_1, S_2) \cdot \hat{t}_1$
F2	= $\bar{h}(S_1, S_2) \cdot \hat{t}_2$
GCØN	= $1/(4d^2)$ factor in $g_1(S_1, S_2)$, ...

G1	$= g_1 (S_1, S_2)$
G2	$= g_2 (S_1, S_2)$
G3	$= g_3 (S_1, S_2)$
G4	$= g_4 (S_1, S_2)$
I1	= do loop index
I2	= do loop index
NINT	= number of steps in S_1 and S_2 used in approximating the integrals for E_1, E_2, ...
SABI	= y component of $\hat{i}$
SALPI	= z component of $\hat{i}$
S1	$= S_1$
S2	$= S_2$
S2X	= initial value of S_2
TPI	$= 2\pi$
T1XI T1YI T1ZI	= x, y and z components of $\hat{t}_1$
T2XI T2YI T2ZI	= x, y and z components of $\hat{t}_2$
XA	= area of each of the four patches
XI	= x coordinate of the center of the connected segment
XS	= x component of $\overline{\rho} (S_1, S_2)$
XSS	= initial x coordinate of $\overline{\rho} (S_1, S_2)$
X1	= x component of $\overline{\rho}$ (d, d) used as reference for computing $\overline{\rho} (S_1, S_2)$
YI	= y coordinate of the center of the connected segment
YS	= y component of $\overline{\rho} (S_1, S_2)$
YSS	= initial y component of $\overline{\rho} (S_1, S_2)$
Y1	= y component of $\overline{\rho}$ (d, d)
ZI	= z coordinate of the center of the connected segment
ZS	= z component of $\overline{\rho} (S_1, S_2)$
ZSS	= initial z component of $\overline{\rho} (S_1, S_2)$
Z1	= z component of $\overline{\rho}$ (d, d)

REFLC

PURPOSE: To generate segment data for structures having plane or cylindrical symmetry by forming symmetric images of a previously defined structure unit.

MODIFICATION: Following each section of the code that operates on segments, a section has been added to perform the same operation on patches. This involves reflecting or rotating the location of the patch center and the vectors $\hat{t}_1$ and $\hat{t}_2$. In addition, each time that a patch is reflected in a coordinate plane the value of SALP for this patch, which has unit magnitude, is changed in sign. A value of +1 for SALP indicates that $(\hat{t}_1, \hat{t}_2, \hat{n})$ are related by $\hat{t}_1 \times \hat{t}_2 = \hat{n}$ while -1 indicates that $\hat{t}_1 \times \hat{t}_2 = -\hat{n}$. All patches generated by subroutine PATCH have SALP = +1.

RE46 to RE65	reflects patches along z axis (about x, y plane).
RE90 to RE109	reflects patches along the y axis.
RE133 to RE152	reflects patches along the x axis.
RE181 to RE204	rotates patches about the z axis to form a structure having cylindrical symmetry.

SOLVES

PURPOSE to control the solution of the matrix equation GI = E where the structure matrix G has been factored into a lower and upper triangular matrix, E is the structure excitation vector, and I is the unknown current vector. For the symmetric structure case, the solution is generated using uncoupled mode solutions.

MODIFICATIONS: The structure excitation vector which is the input B array in SOLVES is filled in subroutine ETMNS. Wire excitation parameters are filled from B(1) to B(N) and surface excitation parameters from B(N + 1) to B(N + 2M) where N and M are the total number of segments and patches respectively. This corresponds to the proper arrangement of the array for an unsymmetric structure as can be seen by the matrix form in figure 9. For the case of a symmetric hybrid structure, however, the arrangement must correspond to the arrangement of equations as shown in figure 9; that is, the wire parameters followed by the surface parameters for each symmetric section. The required rearrangement is performed by the new coding inserted from SS15 to SS33. On the other hand, the solution vector which is returned to the calling program in the B array is expected to be arranged with the wire currents from B(1) to B(N) and the surface currents from B(N + 1) to B(N + 2M). Therefore, the ordering discussed above for the symmetric case must be reversed before control is returned to the calling program. This reordering is performed by the coding inserted from SS89 to SS107.

Two new variables have been defined; NPEQ which is equal to the number of unknowns in a symmetric section and NEQ which is equal to the total number of unknowns. For the case of no symmetry NPEQ = NEQ. The variable NPEQ takes over the function of the NP variable in SOLVES. This replacement has been made at SS39, SS41, SS48, SS60, SS63, SS64, SS67, SS75, SS77, and SS84.

UDOTES

PURPOSE: to compute the E field along wires due to surface currents and to fill the corresponding matrix elements.

METHOD: The column range in the transposed structure matrix between which matrix elements are to be filled is passed to the routine through the parameters I1 and I2. The outer most loop in the routine is a loop over wire observation points which lie within this range. Internal to this is a loop over all the source patches. The source loop is actually composed of two loops; the first loop is over symmetric sections, and the second is over patches in a symmetric section. The electric field due to a surface patch is computed by calling the routine UNERE, and the component of the field tangent to the wire is computed by doting with $\overline{u}_i = u_{xi}\hat{x} + u_{yi}\hat{y} + u_{zi}\hat{z}$, the segment direction. The result is added to the appropriate matrix element. For the case when the observation segment is connected to the surface (indicated by an ICON parameter greater than 10000), the electric field due to the four patches around its base are calculated by the routine PCINT rather than UNERE. As discussed in the theory section, a special interpolation function is used for the current on these patches and the fields are calculated by more careful integration. If a ground plane is present, UNERE is called to compute the field of the image patch. The matrix contribution of the image is added to the same matrix element as the source patch.

Summary of the coding is as follows:

UD20-UD25	initialization
UD27-UD80	wire observation loop
UD37-UD44	checking for a connection segment; in addition checking which end is connected
UD46-UD80	loop over symmetric sections
UD47	computing starting row location -1 for filling the symmetric section (see figure 9)

UD48-UD80	loop over patches in a symmetric section
UD59-UD80	ground loop
UD64-UD72	coding for the case of patches at the base of a connection segment
UD75	routine calculating E field of general patch
UD77-UD78	matrix fill for general patches

SYMBOL DICTIONARY:

CAB	= array containing u_x
CABI	= u_{xi}
CM	= array for the transpose structure matrix elements
EMEL	= array containing matrix elements for the interpolation case computed in PCINT
E1X, E1Y, E1Z	= components of E field due to surface current component in the direction $\hat{t}_1$
E2X, E2Y, E2Z	= components of E field due to surface current component in the direction $\hat{t}_2$
FSIGN	= 1 for end 2 of segment connected to surface, -1 for end 1
I	= observation segment number
ICGO	= parameter used in filling the matrix elements of patches at a connection segment base
IEND	= last observation segment in the input column range
IL	= row location corresponding to the connection segment as a source, this location is needed because of the patch interpolation function
IP	= ground loop index, 1 = free space, 2 = image
IPCH	= storage location of lowest numbered patch at the base of a connecting wire segment, after UD43
I1, I2	= first and last column numbers respectively to be filled in the transposed structure matrix

J	= loop parameter
JL	= row index for matrix element filling
JS	= array location for patch data, after UD49
K	= column number in CM array for filling
L	= index for loop over symmetric sections
NCOL	= number of columns in CM
NOP	= number of symmetric sections
NROW	= number of rows in CM
RFL	= parameter used for image calculations
S	= array containing patch area ($/\lambda^2$)
SAB	= array containing u_y
SABI	= u_{yi}
SALP	= array containing u_z
SALPI	= u_{zi}
T1X, T1Y, T1Z	= arrays containing components of $\hat{t}_1$
T2X, T2Y, T2Z	= arrays containing components of $\hat{t}_2$
XS, YS, ZS	= arrays containing surface patch center point coordinates of patches ($/\lambda$)
ZSEP	= z separation of source and observation point

VARIABLES IN COMMON:

ALP, BET, BI, ICON1, ICON2, ITAG, KSYMP, LD, M, MP, NP, SI, T1XI, T1YI, T1ZI, T2XI, T2YI, T2ZI, X, XI, Y, YI, Z, ZI (other variables in common not referenced in UDOTES, see listing).

UNERE

PURPOSE: to calculate the electric field due to unit currents in the $\hat{t}_1$ and $\hat{t}_2$ directions on a surface patch.

METHOD: The electric field due to a patch j is calculated by the expression

$$\overline{E}(r_o) = \frac{\eta_o}{i8\pi^2}\left[\left(\frac{-1 - i2\pi R/\lambda + 4\pi^2 (R/\lambda)^2}{(R/\lambda)^3}\right)\overline{J}_j + \left(\frac{3 + i6\pi R/\lambda - 4\pi^2 (R/\lambda)^2}{(R/\lambda)^5}\right)\overline{J}_j \cdot (\overline{R}/\lambda)\,(\overline{R}/\lambda)\right] e^{-2\pi R/\lambda}\,\frac{\Delta A_j}{\lambda^2}$$

where $\overline{J}_j = J_{1j}\,\hat{t}_{1j} + J_{2j}\,\hat{t}_{2j}$, $\overline{R}$ is the vector from the source to the observation point, and ΔA_j is the area of the patch. For UNERE, J_{1j} and J_{2j} are unity. The expression above for a single patch is obtained from the surface integral and leading constant in equation 3 where constant current and one step integration are used for the patch.

SYMBOL DICTIONARY:

CONST = $\dfrac{\eta_o}{8\pi^2}$

ER = intermediate complex quantity; $\dfrac{\eta_o}{i8\pi^2}\,e^{-i2\pi R/\lambda}$ at UE15, and Q2 $(\hat{t}_{1j} \cdot (\overline{R}/\lambda))$ at UE18, and Q2 $(\hat{t}_{2j} \cdot (\overline{R}/\lambda))$ at UE22. Q2 is defined below

E1X, E1Y, E1Z = x, y, z components of E field due to J_{1j} for $J_{1j} = 1$

E2X, E2Y, E2Z = x, y, z components of E field due to J_{2j} for $J_{2j} = 1$

Q1 $= \left(\frac{-1 - i2\pi R/\lambda + 4\pi^2 (R/\lambda)^2}{(R/\lambda)^3} \right)$ ER, for ER at UE 15

Q2 $= \left(\frac{3 + i6\pi R/\lambda - 4\pi^2 (R/\lambda)^2}{(R/\lambda)^5} \right)$ ER, for ER at UE15

R $= R/\lambda$

RT $= (R/\lambda)^3$

RX, RY, RZ = x, y, z components of $\overline{R}/\lambda$

R2 $= (R/\lambda)^2$

S $= \Delta A_j$

TPI $= 2\pi$

T1 $= -2\pi R/\lambda$

T1XI, T1YI, T1ZI = x, y, z components of $\hat{t}_{1j}$

T2 $= 4\pi^2 (R/\lambda)^2$

T2XI, T2YI, T2ZI = x, y, z components of $\hat{t}_{2j}$

VARIABLES IN COMMON:

T1XI, T1YI, T1ZI, T2XI, T2YI, T2ZI, XI, YI, ZI

WIRE

PURPOSE: To compute segment coordinates to fill common/DATA/ for a straight line of segments.

MODIFICATION: MP is set equal to M at WI14 since any existing symmetry is destroyed by a new wire.

5.0 ARRAY DIMENSION LIMITATIONS

Limits on the antenna model due to array dimensions are the same in program AMP2 as in program AMP (see reference 3) except for the matrix storage limit and the maximum number of segments and patches. If N is the number of segments in a model and M the number of patches, the minimum array dimensions are as follows:

In-Core Matrix Storage (I_r):

Arrays: COMMON CM(I_r) or AR($2I_r$)
Limit Constant: IRESRV = I_r at MA45 of MAIN

I_r is the amount of core storage available for storage of the interaction matrix elements. I_r must be at least two times (N +2M) and should be as large as possible since a large I_r will reduce the amount of file manipulation required for matrix equation solution and thus can substantially reduce running time. A problem will run in core, without file storage, if the number of complex numbers in the matrix $(N + 2M)^2/L$, where L is the number of symmetric sections, is not greater than I_r.

Maximum Segments and Patches in Model:

Arrays: COMMON/DATA/X(N+M), Y(N+M), Z(N+M), SI(N+M), BI(N+M), ALP(N+M), BET(N+M), ICON1(N+M), ICON2(N+M), ITAG(N+M)

COMMON/ANGL/SALP(N+M)

COMMON/CRNT/AIR(N), AII(N), BIR(N), BII(N), CIR(N), CII(N), CUR(N+3M)

COMMON/SAVE/IX(N+2M), IP(N+2M)

COMMON/ZLOAD/ZARRAY(N)

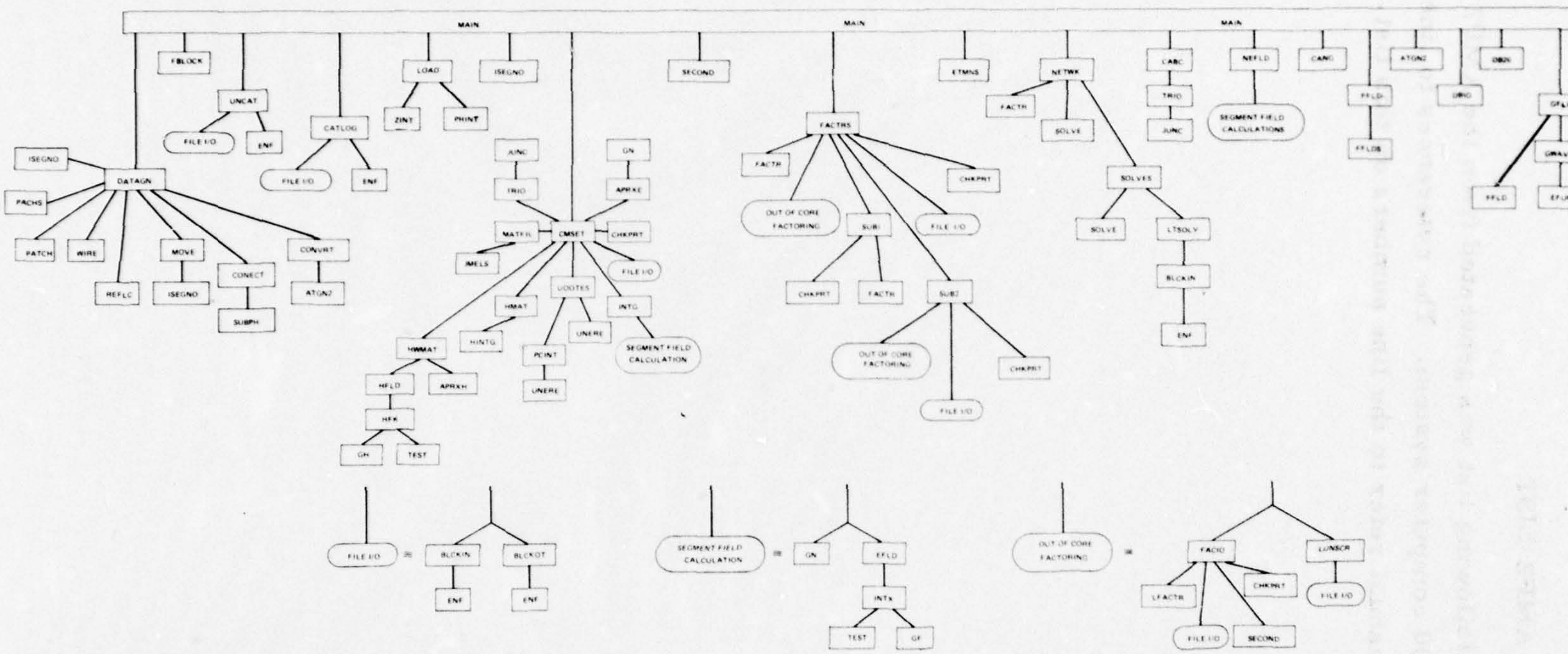
COMMON/SCRATM/D(N+2M) or Y(N+2M)

SUBROUTINE NETWK: RHS (N+2M)

Limit constant: LD = N+M at MA42 of main program

6.0 SUBROUTINE LINKAGE

The following chart shows the organization of subroutines in the AMP2 program. All possible subroutine calls are traced, although in a particular run only certain of the traces will be followed. Routines that are called at more than one point in the program are shown as separate blocks for each call.



AMP2 SUBROUTINE LINKAGE CHART

MBA
0865-13486

7.0 AMP2 LIST

The following list was generated from the AMP2 code which runs on the CDC6600 computer system. The references to line numbers in the text of this manual refer to the line numbers on this list.

```
      PROGRAM AMP2 (INPUT.OUTPUT.TAPE11.TAPE12.TAPE13.TAPE14.TAPE15.TAPE
     116.TAPE17)
      COMMON /DATA/ LD.N.NP.M.MP.X(1000).Y(1000).Z(1000).SI(1000).BI(100
     10).ALP(1000).BET(1000).ICON1(1000).ICON2(1000).ITAG(1000).WLAM.IPS
     2YM
      COMMON CM(4000)
      COMMON /MATPAR/ ICASE.NBLOKS.NPBLK.NLAST.NBLSYM.NPSYM.NLSYM
      COMMON /SAVE/ IX(1500).IP(1500)
      COMMON /RESTRT/ IC1.IC2.IC3.NRES.NPRES.IBLCK.IDUMP.TMDUM.EXTIM
      COMMON /ANGL/ SALP(1000)
      COMMON /JUNK/ NCOX.JOX(25).NCIX.JIX(25).NCOZ.JOZ(25).NCIZ.JIZ(25)
      COMMON /CRNT/ AIR(1000).AII(1000).BIR(1000).BII(1000).CIR(1000).CI
     1I(1000).CUR(2000)
      COMMON /GND/ ZRATI.ZRATI2.CL.CH.SCRWL.SCRWR.NRADL.KSYMP.IFAR.IPERF
      COMMON /SCRATM/ GAIN(3000)
      COMMON /ZLOAD/ ZARRAY(1000)
      DIMENSION CAB(1). SAB(1)
      DIMENSION ISEG1(30). ISEG2(30). Y11R(30). Y11I(30). Y12R(30). Y12I
     1(30). Y22R(30). Y22I(30). NTYP(30). ISANT(10). VSANT(10)
      DIMENSION LDTYP(15). LDTAG(15). LDTAGF(15). LDTAGT(15). ZLR(15). Z
     1LI(15). ZLC(15). IGTP(4). IGAX(4). IGNTP(10)
      DIMENSION COM(13.5). ITAP(7). ATST(18). ISECN(5). PNET(6). HPOL(3)
      DIMENSION FNORM(200)
      DIMENSION T1X(1). T1Y(1). T1Z(1). T2X(1). T2Y(1). T2Z(1)
      EQUIVALENCE (CAB(1).ALP(1)). (SAB(1).BET(1))
      EQUIVALENCE (T1X.SI). (T1Y.ALP). (T1Z.BET). (T2X.ICON1). (T2Y.ICON
     12). (T2Z.ITAG)
      INTEGER AIN.ATST.PNET.HPOL.HBLK.HCIR.HCLIF
      COMPLEX CM.FJ.VSANT.ETH.EPH.ZRATI.CUR.CURI.ZARRAY.ERD.ZRATI2
      COMPLEX EX.EY.EZ.ZPED
      DATA (ITAP(I).I=1.7)/11.12.13.14.15.16.17/
      DATA ATST/2HCE.2HFR.2HLD.2HGN.2HEX.2HNT.2HXQ.2HNF.2HGD.2HRP.2HCM.2
     1HNX.2HEN.2HTL.2HPT.2HRS.2HKH.2HDP/
      DATA ISECN/5H2.3.9.6H2.3.10.5H2.3.3.3H1.0.3H1.0/
      DATA HPOL/6HLINEAR.5HRIGHT.4HLEFT/.HBLK.HCIR/1H .6HCIRCLE/
      DATA PNET/6H      .2H  .6HSTRAIG.2HHT.6HCROSSE.1HD/
      DATA IGTP/6H    - .6HPOWER .6H- DIRE.6HCTIVE /
      DATA IGAX/6H MAJOR.6H MINOR.6H VERT..6H HOR. /
      DATA IGNTP/6H MAJOR.6H AXIS .6H MINOR.6H AXIS .6H   VER.6HTICAL .6
     1H HORIZ.6HONTAL .6H      .6HTOTAL /
      DATA PI.TA.TD.FJ/3.14159265.1.74532925E-02.57.29578.(0..1.)/
      DATA LOADMX.NSMAX.NETMX/15.10.30/.NORMAX/400/.NORMF/200/.LD/1000/
      CALL SECOND (EXTIM)
      ISTART=0
      IRESRV=4000
1     K=0
2     K=K+1
      IF (K.GT.5) K=5
      READ 140. AIN.(COM(I.K).I=1.13)
      IF (AIN.NE.ATST(16)) GO TO 3
      ISTART=1
      GO TO 1
3     IF (K.GT.1) GO TO 4
      PRINT 141
      PRINT 142
      PRINT 143
4     PRINT 144. (COM(I.K).I=1.13)
      IF (AIN.EQ.ATST(11)) GO TO 2
      IF (AIN.EQ.ATST(1)) GO TO 5
      PRINT 145
      STOP
5     CONTINUE
      MPCNT=0
C
C     SETTING UP GEOMETRIC DATA IN SUB. DATAGN AND PRINTING
C
      CALL DATAGN
      IFLOW=1
      NRES=N
      NPRES=NP
      IF (N.EQ.0) GO TO 7
      PRINT 146
      PRINT 147
      DO 6 I=1.N
      ALPI=ALP(I)*TD
      BETI=BET(I)*TD
      PRINT 148. I.X(I).Y(I).Z(I).SI(I).ALPI.BETI.BI(I).ICON1(I).I.ICON2
     1(I).ITAG(I)
      CALP=COS(ALP(I))
      SALP(I)=SIN(ALP(I))
      CAB(I)=CALP*COS(BET(I))
      SAB(I)=CALP*SIN(BET(I))
      IF (SI(I).GT.1.E-20.AND.BI(I).GT.1.E-20) GO TO 6
      PRINT 149
      STOP
6     CONTINUE
7     IF (M.EQ.0) GO TO 9
      PRINT 202
      J=LD+1
      DO 8 I=1.M
      J=J-1
      TMP1=(T1Y(J)*T2Z(J)-T1Z(J)*T2Y(J))*SALP(J)
      TMP2=(T1Z(J)*T2X(J)-T1X(J)*T2Z(J))*SALP(J)
      TMP3=(T1X(J)*T2Y(J)-T1Y(J)*T2X(J))*SALP(J)
      PRINT 203. I.X(J).Y(J).Z(J).TMP1.TMP2.TMP3.BI(J).T1X(J).T1Y(J).T1Z
```

```
     1(J),T2X(J),T2Y(J),T2Z(J)
8     CONTINUE
C
C     DECIDING WHAT TYPE OF MATRIX SOLUTION WILL BE USED IE. IN-CORE
C     OR OUT-OF-CORE
C
9     NEQ=N+2*M
      NPEQ=NP+2*MP
      NOP=NEQ/NPEQ
      IF (N.EQ.0.OR.M.EQ.0) GO TO 10
      IF (N/NP.EQ.M/MP) GO TO 10
      PRINT 204, N,NP,M,MP
      STOP
10    NROW=NEQ
      NROWS=NPEQ
      CALL FBLOCK (NBLOKS,NPBLK,NLAST,IRESRV,NEQ,NPEQ,INT)
      IF (NEQ.NE.NPEQ) GO TO 12
      IF (INT.EQ.1) GO TO 11
      NCOL=NPEQ
      ICASE=1
      GO TO 15
11    NCOL=2*NPBLK
      ICASE=3
      GO TO 15
12    IF (INT.EQ.1) GO TO 13
      NCOL=NPEQ
      NCOLS=NPEQ
      ICASE=2
      GO TO 15
13    CALL FBLOCK (NBLSYM,NPSYM,NLSYM,IRESRV,NPEQ,NPEQ,INT)
      NCOL=2*NPBLK
      IF (INT.EQ.1) GO TO 14
      NCOLS=NPEQ
      ICASE=4
      GO TO 15
14    NCOLS=2*NPSYM
      ICASE=5
15    CONTINUE
      NBLOKX=NBLOKS
      NPBLKX=NPBLK
      NLASTX=NLAST
      NROWX=NROW
      NCOLX=NCOL
      PRINT 150
C
C     FILE PREPARATION FOR OUT-OF-CORE MATRIX SOLUTION, FILES REWOUND
C     AND ENDFILE WRITTEN,     AND TESTING FOR RESTART
C
      IF (ISTART.EQ.0) GO TO 16
      CALL UNCAT
      GO TO 18
16    IF (ICASE.LT.3) GO TO 18
      DO 17 I=1,7
      NUNIT=ITAP(I)
      REWIND NUNIT
      END FILE NUNIT
      REWIND NUNIT
17    CONTINUE
18    CONTINUE
C     DEFAULT VALUES FOR INPUT PARAMETERS AND FLAGS
      IGO=1
      FMHZS=300.
      FMHZ=300.
      NFRQ=1
      RKH=1.
      NLOAD=0
      KSYMP=1
      IXTYP=0
      NET=0
      NRADL=0
      NEAR=-1
      IPTFLG=-2
      IFAR=-1
      ZRATI=CMPLX(1.,0.)
      IPERF=0
      IPED=0
      IDUMP=0
C
C     MAIN INPUT SECTION - STANDARD READ STSTEMENT - JUMPS TO APPRO-
C     PRIATE SECTION FOR SPECIFIC PARAMETER SET UP
C
19    READ 151, AIN,ITMP1,ITMP2,ITMP3,ITMP4,TMP1,TMP2,TMP3,TMP4,TMP5,TMP
     16
      MPCNT=MPCNT+1
      PRINT 152, MPCNT,AIN,ITMP1,ITMP2,ITMP3,ITMP4,TMP1,TMP2,TMP3,TMP4,T
     1MP5,TMP6
      IF (AIN.EQ.ATST(2)) GO TO 21
      IF (AIN.EQ.ATST(3)) GO TO 23
      IF (AIN.EQ.ATST(4)) GO TO 27
      IF (AIN.EQ.ATST(5)) GO TO 31
      IF (AIN.EQ.ATST(6)) GO TO 35
      IF (AIN.EQ.ATST(14)) GO TO 35
      IF (AIN.EQ.ATST(15)) GO TO 38
      IF (AIN.EQ.ATST(7)) GO TO 46
      IF (AIN.EQ.ATST(8)) GO TO 40
      IF (AIN.EQ.ATST(9)) GO TO 43
      IF (AIN.EQ.ATST(10)) GO TO 45
      IF (AIN.EQ.ATST(17)) GO TO 22
      IF (AIN.EQ.ATST(12)) GO TO 1
```

```
      IF (AIN.EQ.ATST(18)) GO TO 39
      IF (AIN.NE.ATST(13)) GO TO 20
      IF (ITMP1.NE.0) CALL CATLOG
      STOP
20    PRINT 153
      STOP
C
C     FREQUENCY PARAMETERS
C
21    IFRQ=ITMP1
      NFRQ=ITMP2
      IF (NFRQ.EQ.0) NFRQ=1
      FMHZ=TMP1
      DELFRQ=TMP2
      IF (IPED.EQ.1) ZPNORM=0.
      IGO=1
      IFLOW=2
      GO TO 19
C
C     MATRIX INTEGRATION LIMIT
C
22    RKH=TMP1
      IGO=1
      IFLOW=2
      GO TO 19
C
C     LOADING PARAMETERS
C
23    IF (IFLOW.EQ.3) GO TO 24
      NLOAD=0
      IFLOW=3
      IF (IGO.GT.2) IGO=2
      IF (ITMP1.EQ.(-1)) GO TO 19
24    NLOAD=NLOAD+1
      IF (NLOAD.LE.LOADMX) GO TO 25
      PRINT 154
      STOP
25    LDTYP(NLOAD)=ITMP1
      LDTAG(NLOAD)=ITMP2
      IF (ITMP4.EQ.0) ITMP4=ITMP3
      LDTAGF(NLOAD)=ITMP3
      LDTAGT(NLOAD)=ITMP4
      IF (ITMP4.GE.ITMP3) GO TO 26
      PRINT 155, NLOAD,ITMP3,ITMP4
      PRINT 159, ISECN(3)
      STOP
26    ZLR(NLOAD)=TMP1
      ZLI(NLOAD)=TMP2
      ZLC(NLOAD)=TMP3
      GO TO 19
C
C     GROUND PARAMETERS UNDER THE ANTENNA
C
27    IFLOW=4
      IF (IGO.GT.2) IGO=2
      IF (ITMP1.NE.(-1)) GO TO 28
      KSYMP=1
      NRADL=0
      GO TO 19
28    IPERF=ITMP1
      IF (M.EQ.0.OR.IPERF.NE.0) GO TO 29
      PRINT 208
      STOP
29    NRADL=ITMP2
      KSYMP=2
      EPSR=TMP1
      SIG=TMP2
      IF (NRADL.EQ.0) GO TO 30
      SCRWLT=TMP3
      SCRWRT=TMP4
      GO TO 19
30    EPSR2=TMP3
      SIG2=TMP4
      CLT=TMP5
      CHT=TMP6
      GO TO 19
C
C     EXCITATION PARAMETERS
C
31    IF (IFLOW.EQ.5) GO TO 32
      IPTFLG=-2
      NSANT=0
      IPED=0
      IFLOW=5
      IF (IGO.GT.3) IGO=3
32    MASYM=ITMP4/10
      IF (ITMP1.GT.0) GO TO 34
      IXTYP=ITMP1
      NTSOL=0
      NSANT=NSANT+1
      IF (NSANT.LE.NSMAX) GO TO 33
      PRINT 156
      STOP
33    ISANT(NSANT)=ISEGNO(ITMP2,ITMP3)
      IPED=ITMP4-MASYM*10
      VSANT(NSANT)=CMPLX(TMP1,TMP2)
      IF (CABS(VSANT(NSANT)).LT.1.E-20) VSANT(NSANT)=(1.,0.)
      ZPNORM=TMP3
      IF (IPED.EQ.1.AND.ZPNORM.GT.0) IPED=2
```

```
      GO TO 19
34    IF (IXTYP.EQ.0) NTSOL=0
      IXTYP=ITMP1
      NTHI=ITMP2
      NPHI=ITMP3
      XPR1=TMP1
      XPR2=TMP2
      XPR3=TMP3
      XPR4=TMP4
      XPR5=TMP5
      XPR6=TMP6
      NSANT=0
      THETIS=XPR1
      PHISS=XPR2
      GO TO 19
C
C     NETWORK PARAMETERS
C
35    IF (IFLOW.EQ.6) GO TO 36
      NET=0
      NTSOL=0
      IFLOW=6
      IF (IGO.GT.3) IGO=3
      IF (ITMP2.EQ.(-1)) GO TO 19
36    NET=NET+1
      IF (NET.LE.NETMX) GO TO 37
      PRINT 157
      STOP
37    NTYP(NET)=2
      IF (AIN.EQ.ATST(6)) NTYP(NET)=1
      ISEG1(NET)=ISEGNO(ITMP1,ITMP2)
      ISEG2(NET)=ISEGNO(ITMP3,ITMP4)
      Y11R(NET)=TMP1
      Y11I(NET)=TMP2
      Y12R(NET)=TMP3
      Y12I(NET)=TMP4
      Y22R(NET)=TMP5
      Y22I(NET)=TMP6
      IF (NTYP(NET).EQ.1.OR.TMP1.GT.0.) GO TO 19
      NTYP(NET)=3
      Y11R(NET)=-TMP1
      GO TO 19
C
C     PRINT CONTROL
C
38    IPTFLG=ITMP1
      IPTAG=ITMP2
      IPTAGF=ITMP3
      IPTAGT=ITMP4
      IF (ITMP4.EQ.0) IPTAGT=IPTAGF
      GO TO 19
C
C     AUTOMATIC FILE DUMP PARAMETERS
C
39    IDUMP=1
      TMDUM=TMP1
      GO TO 19
C
C     NEAR FIELD CALCULATION PARAMETERS
C
40    IF (.NOT.(IFLOW.EQ.8.AND.NFRQ.NE.1)) GO TO 41
      PRINT 158
      PRINT 159, ISECN(1)
41    NEAR=ITMP1
      IF (M.EQ.0) GO TO 42
      PRINT 209
      STOP
42    NRX=ITMP2
      NRY=ITMP3
      NRZ=ITMP4
      XNR=TMP1
      YNR=TMP2
      ZNR=TMP3
      DXNR=TMP4
      DYNR=TMP5
      DZNR=TMP6
      IFLOW=8
      IF (NFRQ.NE.1) GO TO 19
      GO TO (50,58,65,86,87), IGO
C
C     GROUND REPRESENTATION
C
43    IF (.NOT.(IFLOW.EQ.9.AND.NFRQ.NE.1)) GO TO 44
      PRINT 159, ISECN(2)
44    EPSR2=TMP1
      SIG2=TMP2
      CLT=TMP3
      CHT=TMP4
      IFLOW=9
      GO TO 19
C
C     STANDARD OBSERVATION ANGLE PARAMETERS
C
45    IFAR=ITMP1
      NTH=ITMP2
      NPH=ITMP3
      IF (NTH.EQ.0) NTH=1
      IF (NPH.EQ.0) NPH=1
      IPD=ITMP4/10
```

```
      IAVP=ITMP4-IPD*10
      INOR=IPD/10
      IPD=IPD-INOR*10
      IAX=INOR/10
      INOR=INOR-IAX*10
      IF (IAX.NE.0) IAX=1
      IF (IPD.NE.0) IPD=1
      IF (NTH.LT.2.OR.NPH.LT.2) IAVP=0
      IF (IFAR.EQ.1) IAVP=0
      THETS=TMP1
      PHIS=TMP2
      DTH=TMP3
      DPH=TMP4
      RFLD=TMP5
      GNOR=TMP6
      IFLOW=10
      GO TO (50,58,65,86,94), IGO
C
C     EXECUTE CARD  -  CALC. INCLUDING RADIATED FIELDS
C
46    IF (IFLOW.EQ.10.AND.ITMP1.EQ.0) GO TO 19
      IF (NFRQ.EQ.1.AND.ITMP1.EQ.0.AND.IFLOW.GT.7) GO TO 19
      IF (ITMP1.NE.0) GO TO 48
      IF (IFLOW.GT.7) GO TO 47
      IFLOW=7
      GO TO 49
47    IFLOW=11
      GO TO 49
48    IFAR=0
      RFLD=0.
      IPD=0
      IAVP=0
      INOR=0
      IAX=0
      NTH=91
      NPH=1
      THETS=0.
      PHIS=0.
      DTH=1.0
      DPH=0.
      IF (ITMP1.EQ.2) PHIS=90.
      IF (ITMP1.NE.3) GO TO 49
      NPH=2
      DPH=90.
49    GO TO (50,58,65,86,94), IGO
C
C     END OF THE MAIN INPUT SECTION
C     BEGINNING OF THE FREQUENCY DO LOOP
C
50    MHZ=1
51    IF (MHZ.EQ.1) GO TO 53
      IF (IFRQ.EQ.1) GO TO 52
      FMHZ=FMHZ+DELFRQ
      GO TO 53
52    FMHZ=FMHZ*DELFRQ
53    FR=FMHZ/FMHZS
      WLAM=300./FMHZ
      PRINT 160, FMHZ,WLAM
      PRINT 205, RKH
C     FREQUENCY SCALING OF GEOMETRIC PARAMETERS
      FMHZS=FMHZ
      IF (N.EQ.0) GO TO 55
      DO 54 I=1,N
      X(I)=X(I)*FR
      Y(I)=Y(I)*FR
      Z(I)=Z(I)*FR
      SI(I)=SI(I)*FR
54    BI(I)=BI(I)*FR
55    IF (M.EQ.0) GO TO 57
      FR2=FR*FR
      J=LD+1
      DO 56 I=1,M
      J=J-1
      X(J)=X(J)*FR
      Y(J)=Y(J)*FR
      Z(J)=Z(J)*FR
56    BI(J)=BI(J)*FR2
57    IGO=2
C     STRUCTURE SEGMENT LOADING
58    PRINT 161
      IF (NLOAD.NE.0) CALL LOAD (LDTYP,LDTAG,LDTAGF,LDTAGT,ZLR,ZLI,ZLC,N
     1LOAD)
      IF (NLOAD.EQ.0) PRINT 162
C     GROUND PARAMETER
      PRINT 163
      IF (KSYMP.EQ.1) GO TO 61
      IF (IPERF.EQ.1) GO TO 60
      ZRATI=CSQRT(1./(EPSR-SIG*WLAM*59.92*FJ))
      IF (NRADL.EQ.0) GO TO 59
      SCRWL=SCRWLT/WLAM
      SCRWR=SCRWRT/WLAM
      PRINT 185, NRADL,SCRWLT,SCRWRT
      PRINT 164
59    PRINT 165, EPSR,SIG
      GO TO 62
60    PRINT 166
      GO TO 62
61    PRINT 167
62    CONTINUE
```

```
C
C     STRUCTURE MATRIX SET UP
C
      IF (ISTART.NE.0) GO TO 63
      IC1=0
      IC2=IC1
      IC3=IC1
63    NROW=NROWX
      NCOL=NCOLX
      NBLOKS=NBLOKX
      NPBLK=NPBLKX
      NLAST=NLASTX
      CALL SECOND (TIM1)
C     *
      CALL CMSET (NROW,NCOL,CM,NLOAD,RKH)
C     *
      CALL SECOND (TIM2)
      TIM=TIM2-TIM1
C
C     MATRIX FACTORIAZTION
C
      CALL FACTRS (NPEQ,NOP,CM,IP,IX,NROW,NCOL,NCOLS,IPSYM)
      ISTART=0
      IF (ICASE.LE.3) GO TO 64
      NROW=NPEQ
      NCOL=NCOLS
64    CALL SECOND (TIM1)
      TIM2=TIM1-TIM2
      PRINT 168, TIM,TIM2
      IGO=3
      NTSOL=0
C
C     EXCITATION SET UP (RIGHT HAND SIDE, -E INC.)
C
65    NTHIC=1
      NPHIC=1
      INC=1
      NPRINT=0
66    IF (IXTYP.EQ.0) GO TO 68
      IF (IPTFLG.LE.0.OR.IXTYP.EQ.4) PRINT 169
      TMP5=TA*XPR5
      TMP4=TA*XPR4
      IF (IXTYP.LE.3) GO TO 67
      TMP1=XPR1/WLAM
      TMP2=XPR2/WLAM
      TMP3=XPR3/WLAM
      TMP6=XPR6/(WLAM*WLAM)
      PRINT 171, XPR1,XPR2,XPR3,XPR4,XPR5,XPR6
      GO TO 68
67    TMP1=TA*XPR1
      TMP2=TA*XPR2
      TMP3=TA*XPR3
      TMP6=XPR6
      IF (IPTFLG.LE.0) PRINT 170, XPR1,XPR2,XPR3,HPOL(IXTYP),XPR6
68    CALL ETMNS (TMP1,TMP2,TMP3,TMP4,TMP5,TMP6,IXTYP,ISANT,VSANT,NSANT,
     1CUR)
C
C     MATRIX SOLVING  (NETWK CALLS SOLVES)
C
      IF (NET.EQ.0.OR.INC.GT.1) GO TO 72
      PRINT 173
      ITMP3=0
      ITMP1=NTYP(1)
      DO 71 I=1,2
      IF (ITMP1.EQ.3) ITMP1=2
      IF (ITMP1.EQ.2) PRINT 174
      IF (ITMP1.EQ.1) PRINT 175
      DO 70 J=1,NET
      ITMP2=NTYP(J)
      IF ((ITMP2/ITMP1).EQ.1) GO TO 69
      ITMP3=ITMP2
      GO TO 70
69    ITMP4=ISEG1(J)
      ITMP5=ISEG2(J)
      IF (ITMP2.GE.2.AND.Y11I(J).LE.0.) Y11I(J)=WLAM*SQRT((X(ITMP5)-X(IT
     1MP4))**2+(Y(ITMP5)-Y(ITMP4))**2+(Z(ITMP5)-Z(ITMP4))**2)
      PRINT 172, ITAG(ITMP4),ITMP4,ITAG(ITMP5),ITMP5,Y11R(J),Y11I(J),Y12
     1R(J),Y12I(J),Y22R(J),Y22I(J),PNET(2*ITMP2-1),PNET(2*ITMP2)
70    CONTINUE
      IF (ITMP3.EQ.0) GO TO 72
      ITMP1=ITMP3
71    CONTINUE
72    CONTINUE
      IF (INC.GT.1.AND.IPTFLG.GT.0) NPRINT=1
      CALL NETWK (ISEG1,ISEG2,Y11R,Y11I,Y12R,Y12I,Y22R,Y22I,NET,NTYP,ISA
     1NT,VSANT,NSANT,CM,IP,CUR,NROW,NCOL,IX,PIN,PLOSNT,NPRINT,MASYM,ZPED
     2,NTSOL)
      NTSOL=1
      IF (IPED.EQ.0) GO TO 73
      ITMP1=MHZ+4*(MHZ-1)
      IF (ITMP1.GT.(NORMF-3)) GO TO 73
      FNORM(ITMP1)=REAL(ZPED)
      FNORM(ITMP1+1)=AIMAG(ZPED)
      FNORM(ITMP1+2)=CABS(ZPED)
      FNORM(ITMP1+3)=CANG(ZPED)
      IF (IPED.EQ.2) GO TO 73
      IF (FNORM(ITMP1+2).GT.ZPNORM) ZPNORM=FNORM(ITMP1+2)
73    CONTINUE
C
```

```
C     PRINTING STRUCTURE CURRENTS
C
      IF (N.EQ.0) GO TO 82
      IF (IPTFLG.EQ.(-1)) GO TO 75
      IF (IPTFLG.GT.0) GO TO 74
      PRINT 176
      PRINT 177
      GO TO 75
74    IF (IPTFLG.EQ.3.OR.INC.GT.1) GO TO 75
      PRINT 178, XPR3,HPOL(IXTYP),XPR6
75    PLOSS=0.
      ITMP1=0
      JUMP=IPTFLG+1
      DO 81 I=1,N
      CURI=CUR(I)*WLAM
      CMAG=CABS(CURI)
      PH=CANG(CURI)
      IF (NLOAD.EQ.0) GO TO 76
      IF (ABS(REAL(ZARRAY(I))).LT.1.E-20) GO TO 76
      PLOSS=PLOSS+.5*CMAG*CMAG*REAL(ZARRAY(I))*SI(I)
76    IF (JUMP) 80,81,77
77    IF (IPTAG.EQ.0) GO TO 78
      IF (ITAG(I).NE.IPTAG) GO TO 81
78    ITMP1=ITMP1+1
      IF (ITMP1.LT.IPTAGF.OR.ITMP1.GT.IPTAGT) GO TO 81
      IF (IPTFLG.EQ.0) GO TO 80
      IF (IPTFLG.LT.2.OR.INC.GT.NORMF) GO TO 79
      FNORM(INC)=CMAG
      ISAVE=I
79    IF (IPTFLG.NE.3) PRINT 179, XPR1,XPR2,CMAG,PH,I
      GO TO 81
80    PRINT 180, I,ITAG(I),X(I),Y(I),Z(I),SI(I),CURI,CMAG,PH
81    CONTINUE
82    IF (M.EQ.0) GO TO 84
      PRINT 206
      J=N-1
      ITMP1=LD+1
      DO 83 I=1,M
      J=J+2
      ITMP1=ITMP1-1
      ETH=CUR(J)
      EPH=CUR(J+1)
      EX=ETH*T1X(ITMP1)+EPH*T2X(ITMP1)
      EY=ETH*T1Y(ITMP1)+EPH*T2Y(ITMP1)
      EZ=ETH*T1Z(ITMP1)+EPH*T2Z(ITMP1)
      ETHM=CABS(ETH)
      ETHA=CANG(ETH)
      EPHM=CABS(EPH)
      EPHA=CANG(EPH)
83    PRINT 207, I,X(ITMP1),Y(ITMP1),Z(ITMP1),BI(ITMP1),ETHM,ETHA,EPHM,E
     1PHA,EX,EY,EZ
84    IF (IXTYP.NE.0) GO TO 85
      TMP1=PIN-PLOSNT-PLOSS
      TMP2=100.*TMP1/PIN
      PRINT 181, PIN,TMP1,PLOSS,PLOSNT,TMP2
85    CONTINUE
      IGO=4
      IF (IFLOW.NE.7) GO TO 86
      IF (IXTYP.GT.0.AND.IXTYP.LT.4) GO TO 128
      IF (NFRQ.NE.1) GO TO 135
      PRINT 150
      GO TO 19
C
C     CALCULATION OF A,B,C IN CURRENT EXPANSION
C
86    CALL CABC
      IGO=5
C
C     NEAR FIELD CALCULATION
C
87    IF (NEAR.EQ.(-1)) GO TO 93
      PRINT 182
      ZNRT=ZNR-DZNR
      DO 92 I=1,NRZ
      ZNRT=ZNRT+DZNR
      IF (NEAR.EQ.0) GO TO 88
      CTH=COS(TA*ZNRT)
      STH=SIN(TA*ZNRT)
88    YNRT=YNR-DYNR
      DO 92 J=1,NRY
      YNRT=YNRT+DYNR
      IF (NEAR.EQ.0) GO TO 89
      CPH=COS(TA*YNRT)
      SPH=SIN(TA*YNRT)
89    XNRT=XNR-DXNR
      DO 92 KK=1,NRX
      XNRT=XNRT+DXNR
      IF (NEAR.EQ.0) GO TO 90
      XOB=XNRT*STH*CPH
      YOB=XNRT*STH*SPH
      ZOB=XNRT*CTH
      GO TO 91
90    XOB=XNRT
      YOB=YNRT
      ZOB=ZNRT
91    TMP1=XOB/WLAM
      TMP2=YOB/WLAM
      TMP3=ZOB/WLAM
      CALL NEFLD (TMP1,TMP2,TMP3,EX,EY,EZ)
```

```
      TMP1=CABS(EX)
      TMP2=CANG(EX)
      TMP3=CABS(EY)
      TMP4=CANG(EY)
      TMP5=CABS(EZ)
      TMP6=CANG(EZ)
      PRINT 183, XOB,YOB,ZOB,TMP1,TMP2,TMP3,TMP4,TMP5,TMP6
92    CONTINUE
      IF (MHZ.EQ.NFRQ) NEAR=-1
      IF (NFRQ.NE.1) GO TO 93
      PRINT 150
      GO TO 19
93    CONTINUE
C
C     STANDARD FAR FIELD CALCULATION
C
94    IF (IFAR.EQ.-1) GO TO 128
      IF (IFAR.LT.2) GO TO 96
      PRINT 184
      IF (IFAR.LE.3) GO TO 95
      PRINT 185, NRADL,SCRWLT,SCRWRT
      IF (IFAR.EQ.4) GO TO 96
95    IF (IFAR.EQ.2.OR.IFAR.EQ.5) HCLIF=HPOL(1)
      IF (IFAR.EQ.3.OR.IFAR.EQ.6) HCLIF=HCIR
      CL=CLT/WLAM
      CH=CHT/WLAM
      ZRATI2=CSQRT(1./(EPSR2-SIG2*WLAM*59.92*FJ))
      PRINT 186, HCLIF,CLT,CHT,EPSR2,SIG2
96    IF (IFAR.NE.1) GO TO 97
      PRINT 190
      GO TO 99
97    I=2*IPD+1
      J=I+1
      ITMP1=2*IAX+1
      ITMP2=ITMP1+1
      PRINT 187
      IF (RFLD.LT.1.E-20) GO TO 98
      EXRM=WLAM/RFLD
      EXRA=RFLD/WLAM
      EXRA=-360.*(EXRA-AINT(EXRA))
      PRINT 188, RFLD,EXRM,EXRA
98    PRINT 189, IGTP(I),IGTP(J),IGAX(ITMP1),IGAX(ITMP2)
99    IF (IXTYP.EQ.0) GO TO 101
      IF (IXTYP.EQ.4) GO TO 100
      PRAD=0.
      GCON=4.*PI/(1.+XPR6*XPR6)
      GCOP=GCON
      GO TO 102
100   PIN=394.51*XPR6*XPR6*WLAM*WLAM
101   GCOP=WLAM*WLAM*2.*PI/(376.73*PIN)
      PRAD=PIN-PLOSS-PLOSNT
      GCON=GCOP
      IF (IPD.NE.0) GCON=GCON*PIN/PRAD
102   I=0
      GMAX=-1.E10
      PINT=0.
      TMP1=DPH*TA
      TMP2=.5*DTH*TA
      PHI=PHIS-DPH
      DO 123 KPH=1,NPH
      PHI=PHI+DPH
      PHA=PHI*TA
      THET=THETS-DTH
      DO 123 KTH=1,NTH
      THET=THET+DTH
      IF (KSYMP.EQ.2.AND.THET.GT.90.01.AND.IFAR.NE.1) GO TO 123
      THA=THET*TA
      IF (IFAR.EQ.1) GO TO 103
      CALL FFLD (THA,PHA,ETH,EPH)
      GO TO 104
103   CALL GFLD (RFLD/WLAM,PHA,THET/WLAM,ETH,EPH,ERD,ZRATI,KSYMP)
      ERDM=CABS(ERD)
      ERDA=CANG(ERD)
104   ETHM2=REAL(ETH*CONJG(ETH))
      ETHM=SQRT(ETHM2)
      ETHA=CANG(ETH)
      EPHM2=REAL(EPH*CONJG(EPH))
      EPHM=SQRT(EPHM2)
      EPHA=CANG(EPH)
      IF (IFAR.EQ.1) GO TO 122
C     ELLIPTICAL POLARIZATION CALC.
      IF (ETHM2.GT.1.E-20.OR.EPHM2.GT.1.E-20) GO TO 105
      TILTA=0.
      EMAJR2=0.
      EMINR2=0.
      AXRAT=0.
      ISENS=HBLK
      GO TO 110
105   DFAZ=EPHA-ETHA
      IF (EPHA.LT.0.) GO TO 106
      DFAZ2=DFAZ-360.
      GO TO 107
106   DFAZ2=DFAZ+360.
107   IF (ABS(DFAZ).GT.ABS(DFAZ2)) DFAZ=DFAZ2
      CDFAZ=COS(DFAZ*TA)
      TSTOR1=ETHM2-EPHM2
      TSTOR2=2.*EPHM*ETHM*CDFAZ
      TILTA=.5*ATGN2(TSTOR2,TSTOR1)
      STILTA=SIN(TILTA)
```

```
      TSTOR1=TSTOR1*STILTA*STILTA
      TSTOR2=TSTOR2*STILTA*COS(TILTA)
      EMAJR2=-TSTOR1+TSTOR2+ETHM2
      EMINR2=TSTOR1-TSTOR2+EPHM2
      IF (EMINR2.LT.0.) EMINR2=0.
      AXRAT=SQRT(EMINR2/EMAJR2)
      TILTA=TILTA*TO
      IF (AXRAT.GT.1.E-5) GO TO 108
      ISENS=HPOL(1)
      GO TO 110
108   IF (DFAZ.GT.0.) GO TO 109
      ISENS=HPOL(2)
      GO TO 110
109   ISENS=HPOL(3)
110   GNMJ=DB10(GCON*EMAJR2)
      GNMN=DB10(GCON*EMINR2)
      GNV=DB10(GCON*ETHM2)
      GNH=DB10(GCON*EPHM2)
      GTOT=DB10(GCON*(ETHM2+EPHM2))
      IF (INOR.LT.1) GO TO 117
      I=I+1
      IF (I.GT.NORMAX) GO TO 117
      GO TO (111,112,113,114,115), INOR
111   TSTOR1=GNMJ
      GO TO 116
112   TSTOR1=GNMN
      GO TO 116
113   TSTOR1=GNV
      GO TO 116
114   TSTOR1=GNH
      GO TO 116
115   TSTOR1=GTOT
116   GAIN(I)=TSTOR1
      IF (TSTOR1.GT.GMAX) GMAX=TSTOR1
117   IF (IAVP.EQ.0) GO TO 118
      TSTOR1=GCOP*(ETHM2+EPHM2)
      TMP3=THA-TMP2
      TMP4=THA+TMP2
      IF (KTH.EQ.1) TMP3=THA
      IF (KTH.EQ.NTH) TMP4=THA
      DA=ABS(TMP1*(COS(TMP3)-COS(TMP4)))
      IF (KPH.EQ.1.OR.KPH.EQ.NPH) DA=.5*DA
      PINT=PINT+TSTOR1*DA
      IF (IAVP.EQ.2) GO TO 123
118   IF (IAX.EQ.1) GO TO 119
      TMP5=GNMJ
      TMP6=GNMN
      GO TO 120
119   TMP5=GNV
      TMP6=GNH
120   IF (RFLD.LT.1.E-20) GO TO 121
      ETHM=ETHM*EXRM
      ETHA=ETHA+EXRA
      EPHM=EPHM*EXRM
      EPHA=EPHA+EXRA
121   PRINT 191, THET,PHI,TMP5,TMP6,GTOT,AXRAT,TILTA,ISENS,ETHM,ETHA,EPH
     1M,EPHA
      GO TO 123
122   PRINT 192, RFLD,PHI,THET,ETHM,ETHA,EPHM,EPHA,ERDM,ERDA
123   CONTINUE
      IF (IAVP.EQ.0) GO TO 124
      TMP3=THETS*TA
      TMP4=TMP3+DTH*TA*FLOAT(NTH-1)
      TMP3=ABS(DPH*TA*FLOAT(NPH-1)*(COS(TMP3)-COS(TMP4)))
      PINT=PINT/TMP3
      TMP3=TMP3/PI
      PRINT 193, PINT,TMP3
124   IF (INOR.EQ.0) GO TO 128
      IF (ABS(GNOR).GT.1.E-20) GMAX=GNOR
      ITMP1=(INOR-1)*2+1
      ITMP2=ITMP1+1
      PRINT 194, IGNTP(ITMP1),IGNTP(ITMP2),GMAX
      ITMP2=NPH*NTH
      IF (ITMP2.GT.NORMAX) ITMP2=NORMAX
      ITMP1=(ITMP2+2)/3
      ITMP2=ITMP1*3-ITMP2
      ITMP3=ITMP1
      ITMP4=2*ITMP1
      IF (ITMP2.EQ.2) ITMP4=ITMP4-1
      DO 125 I=1,ITMP1
      ITMP3=ITMP3+1
      ITMP4=ITMP4+1
      J=(I-1)/NTH
      TMP1=THETS+FLOAT(I-J*NTH-1)*DTH
      TMP2=PHIS+FLOAT(J)*DPH
      J=(ITMP3-1)/NTH
      TMP3=THETS+FLOAT(ITMP3-J*NTH-1)*DTH
      TMP4=PHIS+FLOAT(J)*DPH
      J=(ITMP4-1)/NTH
      TMP5=THETS+FLOAT(ITMP4-J*NTH-1)*DTH
      TMP6=PHIS+FLOAT(J)*DPH
      TSTOR1=GAIN(I)-GMAX
      IF (I.EQ.ITMP1.AND.ITMP2.NE.0) GO TO 126
      TSTOR2=GAIN(ITMP3)-GMAX
      PINT=GAIN(ITMP4)-GMAX
125   PRINT 195, TMP1,TMP2,TSTOR1,TMP3,TMP4,TSTOR2,TMP5,TMP6,PINT
      GO TO 128
126   IF (ITMP2.EQ.2) GO TO 127
      TSTOR2=GAIN(ITMP3)-GMAX
```

```
      PRINT 195, TMP1,TMP2,TSTOR1,TMP3,TMP4,TSTOR2
      GO TO 128
127   PRINT 195, TMP1,TMP2,TSTOR1
128   IF (IXTYP.EQ.0.OR.IXTYP.EQ.4) GO TO 134
      NTHIC=NTHIC+1
      INC=INC+1
      XPR1=XPR1+XPR4
      IF (NTHIC.LE.NTHI) GO TO 66
      NTHIC=1
      XPR1=THETIS
      XPR2=XPR2+XPR5
      NPHIC=NPHIC+1
      IF (NPHIC.LE.NPHI) GO TO 66
      NPHIC=1
      XPR2=PHISS
      IF (IPTFLG.LT.2) GO TO 134
      ITMP1=NTHI*NPHI
      IF (ITMP1.LE.NORMF) GO TO 129
      ITMP1=NORMF
      PRINT 196
129   TMP1=FNORM(1)
      DO 130 J=2,ITMP1
      IF (FNORM(J).GT.TMP1) TMP1=FNORM(J)
130   CONTINUE
      PRINT 197, TMP1,XPR3,HPOL(IXTYP),XPR6,ISAVE
      DO 133 J=1,NPHI
      ITMP2=NTHI*(J-1)
      DO 131 I=1,NTHI
      ITMP3=I+ITMP2
      IF (ITMP3.GT.ITMP1) GO TO 132
      TMP2=FNORM(ITMP3)/TMP1
      TMP3=DB20(TMP2)
      PRINT 198, XPR1,XPR2,TMP3,TMP2
      XPR1=XPR1+XPR4
131   CONTINUE
132   XPR1=THETIS
      XPR2=XPR2+XPR5
133   CONTINUE
      XPR2=PHISS
134   IF (MHZ.EQ.NFRQ) IFAR=-1
      IF (NFRQ.NE.1) GO TO 135
      PRINT 150
      GO TO 19
135   MHZ=MHZ+1
      IF (MHZ.LE.NFRQ) GO TO 51
      IF (IPED.EQ.0) GO TO 138
      PRINT 199, ISANT(NSANT),ZPNORM
      ITMP1=NFRQ
      IF (ITMP1.LE.(NORMF/4)) GO TO 136
      ITMP1=NORMF/4
      PRINT 200
136   IF (IFRQ.EQ.0) TMP1=FMHZ-(NFRQ-1)*DELFRQ
      IF (IFRQ.EQ.1) TMP1=FMHZ/(DELFRQ**(NFRQ-1))
      DO 137 I=1,ITMP1
      ITMP2=I+4*(I-1)
      TMP2=FNORM(ITMP2)/ZPNORM
      TMP3=FNORM(ITMP2+1)/ZPNORM
      TMP4=FNORM(ITMP2+2)/ZPNORM
      TMP5=FNORM(ITMP2+3)
      PRINT 201, TMP1,FNORM(ITMP2),FNORM(ITMP2+1),FNORM(ITMP2+2),FNORM(I
     1TMP2+3),TMP2,TMP3,TMP4,TMP5
      IF (IFRQ.EQ.0) TMP1=TMP1+DELFRQ
      IF (IFRQ.EQ.1) TMP1=TMP1*DELFRQ
137   CONTINUE
      PRINT 150
138   CONTINUE
      NFRQ=1
      MHZ=1
      GO TO 19
C
C
140   FORMAT (A2,13A6)
141   FORMAT (1H1)
142   FORMAT (///,33X,30H******************************,//,36X,24HANTENN
     1A MODELING PROGRAM,//,33X,30H******************************)
143   FORMAT (////,37X,24H- - - - COMMENTS - - - -,/)
144   FORMAT (25X,13A6)
145   FORMAT (///,10X,34HINCORRECT LABEL FOR A COMMENT CARD)
146   FORMAT (////33X,33H- - - - SEGMENTATION DATA - - - -,//,40X,21HCOO
     1RDINATES IN METERS,//,25X,50HI+ AND I- INDICATE THE SEGMENTS BEFOR
     2E AND AFTER I,//)
147   FORMAT (2X,4HSEG.,3X,26HCOORDINATES OF SEG. CENTER,5X,4HSEG.,5X,18
     1HORIENTATION ANGLES,4X,4HWIRE,4X,15HCONNECTION DATA,3X,3HTAG,/,2X,
     23HNO.,7X,1HX,9X,1HY,9X,1HZ,7X,6HLENGTH,5X,5HALPHA,5X,4HBETA,6X,6HR
     3ADIUS,4X,2HI-,3X,1HI,4X,2HI+,4X,3HNO.)
148   FORMAT (1X,I5,4F10.5,1X,3F10.5,1X,3I5,2X,I5)
149   FORMAT (19H SEGMENT DATA ERROR)
150   FORMAT (/////)
151   FORMAT (A2,I3,3I5,6E10.3)
152   FORMAT (1X,19H***** DATA CARD NO.,I3,3X,A2,1X,I3,3(1X,I5),6(1X,E12
     1.5))
153   FORMAT (///,10X,45HFAULTY DATA CARD LABEL AFTER GEOMETRY SECTION)
154   FORMAT (///,10X,48HNUMBER OF LOADING CARDS EXCEEDS STORAGE ALLOTTE
     1D)
155   FORMAT (///,10X,31HDATA FAULT ON LOADING CARD NO.=,I5,5X,11HITAG S
     1TEP1=,I5,29H  IS GREATER THAN ITAG STEP2=,I5)
156   FORMAT (///,10X,51HNUMBER OF EXCITATION CARDS EXCEEDS STORAGE ALLO
     1TTED)
157   FORMAT (///,10X,48HNUMBER OF NETWORK CARDS EXCEEDS STORAGE ALLOTTE
```

```
     1D)
158   FORMAT (///,10X,79HWHEN MULTIPLE FREQUENCIES ARE REQUESTED, ONLY O
     1NE NEAR FIELD CARD CAN BE USED -,/,10X,22HLAST CARD READ IS USED)
159   FORMAT (10X,25HSEE USERS MANUAL SECTION ,A6)
160   FORMAT (////,33X,33H- - - - - - FREQUENCY - - - - - -,//,36X,10HFR
     1EQUENCY=,E11.4,4H MHZ,/,36X,11HWAVELENGTH=,E11.4,7H METERS)
161   FORMAT (///,30X,40H - - - STRUCTURE IMPEDANCE LOADING - - -)
162   FORMAT (/,35X,28HTHIS STRUCTURE IS NOT LOADED)
163   FORMAT (///,34X,31H- - - ANTENNA ENVIRONMENT - - -,/)
164   FORMAT (40X,21HMEDIUM UNDER SCREEN -)
165   FORMAT (40X,27HRELATIVE DIELECTRIC CONST.=,F7.3,/,40X,13HCONDUCTIV
     1ITY=,E10.3,11H MHOS/METER)
166   FORMAT (42X,14HPERFECT GROUND)
167   FORMAT (44X,10HFREE SPACE)
168   FORMAT (///,32X,25H- - - MATRIX TIMING - - -,//,24X,5HFILL=,F9.3,1
     15H SEC.,  FACTOR=,F9.3,5H SEC.)
169   FORMAT (///,40X,22H- - - EXCITATION - - -)
170   FORMAT (/,4X,10HPLANE WAVE,4X,6HTHETA=,F7.2,11H DEG,  PHI=,F7.2,11
     1H DEG,  ETA=,F7.2,13H DEG,  TYPE -,A6,15H,  AXIAL RATIO=,F6.3)
171   FORMAT (/,31X,17HPOSITION (METERS),14X,18HORIENTATION (DEG)=/,28X,
     11HX,12X,1HY,12X,1HZ,10X,5HALPHA,5X,4HBETA,4X,13HDIPOLE MOMENT,//,4
     2X,14HCURRENT SOURCE,1X,3(3X,F10.5),1X,2(3X,F7.2),4X,F8.3)
172   FORMAT (4X,4(I5,1X),6(3X,E11.4),3X,A6,A2)
173   FORMAT (///,44X,24H- - - NETWORK DATA - - -)
174   FORMAT (/,6X,18H- FROM -    - TO -,11X,17HTRANSMISSION LINE,15X,36
     1H-  -  SHUNT ADMITTANCES (MHOS)  -  -,14X,4HLINE,/,6X,21HTAG  SEG.
     2  TAG  SEG.,6X,9HIMPEDANCE,6X,6HLENGTH,12X,11H- END ONE -,17X,11H
     3- END TWO -,12X,4HTYPE,/,6X,21HNO.   NO.   NO.   NO.,9X,4HOHMS,8X,
     46HMETERS,9X,4HREAL,10X,5HIMAG.,9X,4HREAL,10X,5HIMAG.)
175   FORMAT (/,6X,8H- FROM -,4X,6H- TO -,26X,45H-  -  ADMITTANCE MATRIX
     1 ELEMENTS (MHOS)  -  -,/,6X,21HTAG  SEG.   TAG  SEG.,13X,9H(ONE,ON
     2E),19X,9H(ONE,TWO),19X,9H(TWO,TWO),/,6X,21HNO.   NO.   NO.   NO.,8
     3X,4HREAL,10X,5HIMAG.,9X,4HREAL,10X,5HIMAG.,9X,4HREAL,10X,5HIMAG.)
176   FORMAT (///,29X,33H- - - CURRENTS AND LOCATION - - -,//,33X,24HDIS
     1TANCES IN WAVELENGTHS)
177   FORMAT (//,2X,4HSEG.,2X,3HTAG,4X,21HCOORD. OF SEG. CENTER,5X,4HSEG
     1.,12X,26H- - - CURRENT (AMPS) - - -,/,2X,3HNO.,3X,3HNO.,5X,1HX,8X,
     21HY,8X,1HZ,6X,6HLENGTH,5X,4HREAL,8X,5HIMAG.,7X,4HMAG.,8X,5HPHASE)
178   FORMAT (///,33X,40H- - - RECEIVING PATTERN PARAMETERS - - -,/,43X,
     14HETA=,F7.2,8H DEGREES,/,43X,6HTYPE -,A6,/,43X,12HAXIAL RATIO=,F6.
     23,//,11X,5HTHETA,6X,3HPHI,10X,13H-  CURRENT  -,9X,3HSEG,/,11X,5H(D
     3EG),5X,5H(DEG),7X,9HMAGNITUDE,4X,5HPHASE,6X,3HNO.,/)
179   FORMAT (10X,2(F7.2,3X),1X,E11.4,3X,F7.2,4X,I5)
180   FORMAT (1X,2I5,3F9.4,F9.5,1X,3E12.4,F9.3)
181   FORMAT (///,40X,24H- - - POWER BUDGET - - -,//,43X,15HINPUT POWER
     1  =,E11.4,6H WATTS,/,43X,15HRADIATED POWER=,E11.4,6H WATTS,/,43X,1
     25HSTRUCTURE LOSS=,E11.4,6H WATTS,/,43X,15HNETWORK LOSS  =,E11.4,6H
     3 WATTS,/,43X,15HEFFICIENCY    =,F7.2,8H PERCENT)
182   FORMAT (///,39X,23H- - - NEAR FIELDS - - -,//,12X,14H-  LOCATION
     1-,21X,8H-  EX  -,15X,8H-  EY  -,15X,8H-  EZ  -,/,9X,1HX,10X,1HY,10
     2X,1HZ,10X,9HMAGNITUDE,3X,5HPHASE,6X,9HMAGNITUDE,3X,5HPHASE,6X,9HMA
     3GNITUDE,3X,5HPHASE,/,6X,6HMETERS,5X,6HMETERS,5X,6HMETERS,8X,7HVOLT
     4S/M,3X,7HDEGREES,6X,7HVOLTS/M,3X,7HDEGREES,6X,7HVOLTS/M,3X,7HDEGRE
     5ES)
183   FORMAT (2X,3(2X,F9.4),1X,3(3X,E11.4,2X,F7.2))
184   FORMAT (///,31X,39H- - - FAR FIELD GROUND PARAMETERS - - -,//)
185   FORMAT (40X,25HRADIAL WIRE GROUND SCREEN,/,40X,I5,6H WIRES,/,40X,1
     12HWIRE LENGTH=,F8.2,7H METERS,/,40X,12HWIRE RADIUS=,E10.3,7H METER
     2S)
186   FORMAT (40X,A6,6H CLIFF,/,40X,14HEDGE DISTANCE=,F9.2,7H METERS,/,4
     10X,7HHEIGHT=,F8.2,7H METERS,/,40X,15HSECOND MEDIUM -,/,40X,27HRELA
     2TIVE DIELECTRIC CONST.=,F7.3,/,40X,13HCONDUCTIVITY=,E10.3,5H MHOS)
187   FORMAT (///,48X,30H- - - RADIATION PATTERNS - - -,/)
188   FORMAT (54X,6HRANGE=,E13.6,7H METERS,/,54X,16H2.*PI*EXP(-JKR)/,5H(
     1KR)=,E12.5,9H AT PHASE,F7.2,8H DEGREES,/)
189   FORMAT (4X,14H- - ANGLES - -,8X,2A6,7HGAINS -,7X,24H- - - POLARIZA
     1TION - - -,4X,20H- - - E(THETA) - - -,4X,18H- - - E(PHI) - - -,/,
     24X,5HTHETA,5X,3HPHI,7X,A6,3X,A6,4X,5HTOTAL,6X,5HAXIAL,5X,4HTILT,3X
     3,5HSENSE,2(5X,9HMAGNITUDE,4X,6HPHASE ),/,1X,2(2X,7HDEGREES),3(7X,
     42HDB),8X,5HRATIO,5X,4HDEG.,8X,2(6X,7HVOLTS/M,4X,7HDEGREES))
190   FORMAT (///,28X,40H - - - RADIATED FIELDS NEAR GROUND - - -,//,8X,
     120H- - - LOCATION - - -,10X,16H- - E(THETA) - -,8X,14H- - E(PHI) -
     2 -,8X,17H- - E(RADIAL) - -,/,7X,3HRHO,6X,3HPHI,9X,1HZ,12X,3HMAG,6X
     3,5HPHASE,9X,3HMAG,6X,5HPHASE,9X,3HMAG,6X,5HPHASE,/,5X,6HMETERS,3X,
     47HDEGREES,4X,6HMETERS,8X,7HVOLTS/M,3X,7HDEGREES,6X,7HVOLTS/M,3X,7H
     5DEGREES,6X,7HVOLTS/M,3X,7HDEGREES,/)
191   FORMAT (1X,2F9.2,2X,3F9.2,2X,F9.5,F9.2,2X,A6,2(3X,E12.5,F9.2))
192   FORMAT (3X,F9.2,2X,F7.2,2X,F9.2,1X,3(3X,E11.4,2X,F7.2))
193   FORMAT (  //,3X,19HAVERAGE POWER GAIN=,E12.5,7X,16HSOLID ANGLE USE
     1D,15H IN AVERAGING=(,F7.4,16H)*PI STERADIANS.,//)
194   FORMAT (//,37X,31H- - - - NORMALIZED GAIN - - - -,//,37X,2A6,4HGAI
     1N,/,38X,22HNORMALIZATION FACTOR =,F9.2,3H DB,//,3(4X,14H- - ANGLES
     2 - -,6X,4HGAIN,7X),/,3(4X,5HTHETA,5X,3HPHI,8X,2HDB,8X),/,3(3X,7HDE
     3GREES,2X,7HDEGREES,16X))
195   FORMAT (3(1X,2F9.2,1X,F9.2,6X))
196   FORMAT (///,4X,51HRECEIVING PATTERN STORAGE TOO SMALL,ARRAY TRUNCA
     1TED)
197   FORMAT (///,32X,40H- - - NORMALIZED RECEIVING PATTERN - - -,/,41X,
     121HNORMALIZATION FACTOR=,E11.4,/,41X,4HETA=,F7.2,8H DEGREES,/,41X,
     26HTYPE -,A6,/,41X,12HAXIAL RATIO=,F6.3,/,41X,12HSEGMENT NO.=,I5,//
     3,21X,5HTHETA,6X,3HPHI,9X,13H-  PATTERN  -,/,21X,5H(DEG),5X,5H(DEG)
     4,8X,2HDB,8X,9HMAGNITUDE,/)
198   FORMAT (20X,2(F7.2,3X),1X,F7.2,4X,E11.4)
199   FORMAT (///,36X,32H- - - INPUT IMPEDANCE DATA - - -,/,45X,18HSOURC
     1E SEGMENT NO.,I4,/,45X,21HNORMALIZATION FACTOR=,E12.5,//,7X,5HFREQ
     2.,13X,34H-  -  UNNORMALIZED IMPEDANCE  -  -,21X,32H-  -  NORMALIZE
     3D IMPEDANCE  -  -,/,19X,10HRESISTANCE,4X,9HREACTANCE,6X,9HMAGNITUD
     4E,4X,5HPHASE,7X,10HRESISTANCE,4X,9HREACTANCE,6X,9HMAGNITUDE,4X,5HP
     5HASE,/,8X,3HMHZ,11X,4HOHMS,10X,4HOHMS,11X,4HOHMS,5X,7HDEGREES,47X,
```

```
     67HDEGREES,/)
 200  FORMAT (///,4X,62HSTORAGE FOR IMPEDANCE NORMALIZATION TOO SMALL, A
     1RRAY TRUNCATED)
 201  FORMAT (4X,F9.3,2X,2(2X,E12.5),3X,E12.5,2X,F7.2,2X,2(2X,E12.5),3X,
     1E12.5,2X,F7.2)
 202  FORMAT (////,44X,30H- - - SURFACE PATCH DATA - - -,//,49X,21HCOORD
     1INATES IN METERS,//,1X,5HPATCH,3X,27HCOORDINATES OF PATCH CENTER,8
     2X,18HUNIT NORMAL VECTOR,9X,5HPATCH,12X,34HCOMPONENTS OF UNIT TANGE
     3NT VECTORS,/,3X,3HNO.,6X,1HX,9X,1HY,9X,1HZ,11X,1HX,8X,1HY,8X,1HZ,9
     4X,4HAREA,7X,2HX1,6X,2HY1,6X,2HZ1,8X,2HX2,6X,2HY2,6X,2HZ2)
 203  FORMAT (1X,I5,3F10.5,2X,3F9.4,F11.5,2X,3F8.4,2X,3F8.4)
 204  FORMAT (40H GEOMETRY DATA INCONSISTENT - N,NP,M,MP=,4I6)
 205  FORMAT (////,20X,55HAPPROXIMATE INTEGRATION EMPLOYED FOR SEGMENTS
     1MORE THAN,F8.3,18H WAVELENGTHS APART)
 206  FORMAT (////,45X,38H- - - - SURFACE PATCH CURRENTS - - - -,//,54X,
     123HDISTANCE IN WAVELENGTHS,/,54X,21HCURRENT IN AMPS/METER,//,40X,2
     26H- - SURFACE COMPONENTS - -,19X,34H- - - RECTANGULAR COMPONENTS -
     3 - -,/,1X,5HPATCH,4X,12HPATCH CENTER,6X,5HPATCH,3X,16HTANGENT VECT
     4OR 1,3X,16HTANGENT VECTOR 2,11X,1HX,19X,1HY,19X,1HZ,/,2X,3HNO.,4X,
     51HX,6X,1HY,6X,1HZ,5X,4HAREA,4X,4HMAG.,7X,5HPHASE,3X,4HMAG.,7X,5HPH
     *ASE,3(4X,4HREAL,6X,6HIMAG. ))
 207  FORMAT (1X,I4,3F7.3,F8.5,2(E11.4,F8.2),6E10.2)
 208  FORMAT (59H ERROR--ONLY PERFECT GROUND IS ALLOWED WITH SURFACE PAT
     1CHES)
 209  FORMAT (71H ERROR--NEAR FIELD MAY NOT BE COMPUTED WHEN SURFACE PAT
     1CHES ARE PRESENT)
      END


      SUBROUTINE APRXE (R,S,ETA,PI2,DIJ,DIR,ZP,RH,ETR,ETI,IP)
C
C     APRXE CALCULATES THE ELECTRIC FIELD OF AN INFINITESIMAL CURRENT
C     ELEMENT FOR THE MATRIX ELEMENT APPROXIMATION FOR SEPARATION
C     DISTANCES GREATER THAN RKH.
C
      DIMENSION EZE(2), EPE(2)
      COMPLEX C1,ER,ET,EZ,EP
      EQUIVALENCE (EZ,EZE), (EP,EPE)
      RKH1=PI2*R
      A0=ZP/R
      A1=RH/R
      C1=CMPLX(COS(RKH1),-SIN(RKH1))
      ER=S*ETA*A0*C1*CMPLX(1.,-1./RKH1)/(PI2*R*R)
      ET=S*ETA*A1*C1*CMPLX(1.,RKH1-1./RKH1)/(2.*PI2*R*R)
      EZ=ER*A0-ET*A1
      EP=ER*A1+ET*A0
      IF (IP.EQ.2) CALL GN (EZE(1),EZE(2),EPE(1),EPE(2))
      ET=DIJ*EZ+DIR*EP
      ETR=ETR+REAL(ET)
      ETI=ETI+AIMAG(ET)
      RETURN
      END


      SUBROUTINE APRXH (R,R2,S,TPI,ST,TWHR,TWHI,ILC,PX,PY,PZ,RFL)
C
C     APRXH CALCULATES THE MAGNETIC FIELD OF AN INFINITESIMAL CURRENT
C     ELEMENT FOR THE MATRIX ELEMENT APPROXIMATION FOR SEPARATION
C     DISTANCES GREATER THAN RKH.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      DIMENSION TWHR(3,2), TWHI(3,2)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)
      COMPLEX H
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      RK=TPI*R
      H=S*ST*CMPLX(COS(RK),-SIN(RK))*CMPLX(1./R2,TPI/R)/(2.*TPI)*RFL
      DO 1 K=1,2
      IF (K.EQ.1) PDT=PX*T2X(ILC)+PY*T2Y(ILC)+PZ*T2Z(ILC)
      IF (K.EQ.2) PDT=PX*T1X(ILC)+PY*T1Y(ILC)+PZ*T1Z(ILC)
      PDT=PDT*SALP(ILC)
      TWHR(2,K)=TWHR(2,K)-PDT*REAL(H)
      TWHI(2,K)=TWHI(2,K)-PDT*AIMAG(H)
 1    CONTINUE
      RETURN
      END


      FUNCTION ATGN2 (X,Y)
C
C     ATGN2 IS ARCTANGENT FUNCTION MODIFIED TO RETURN 0. WHEN X=Y=0.
C
      IF (X) 3,1,3
 1    IF (Y) 3,2,3
 2    ATGN2=0.
      RETURN
 3    ATGN2=ATAN2(X,Y)
      RETURN
      END
```

```
      SUBROUTINE BLCKOT (NUNIT,I1,I2,NBLKS,NEOF)
C
C     BLCKOT CONTROLS THE READING AND WRITING OF MATRIX BLOCKS ON FILES
C     FOR THE OUT-OF-CORE MATRIX SOLUTION.
C
      COMMON AR(8000)
      LOGICAL ENF
1     WRITE (NUNIT) (AR(J),J=I1,I2)
      RETURN
      ENTRY BLCKIN
      DO 2 I=1,NBLKS
      READ (NUNIT) (AR(J),J=I1,I2)
      IF (ENF(NUNIT)) GO TO 3
2     CONTINUE
      RETURN
3     PRINT 4, NUNIT,NBLKS,NEOF
      IF (NEOF.NE.777) STOP
      NEOF=0
      RETURN
C
4     FORMAT (13H  EOF ON UNIT,I3,9H  NBLKS= ,I3,8H  NEOF= ,I5)
      END


      SUBROUTINE CABC
C
C     CABC COMPUTES COEFFICIENTS OF THE CONSTANT (A), SINE (B), AND
C     COSINE (C) TERMS IN THE CURRENT INTERPOLATION FUNCTIONS FOR THE
C     CURRENT VECTOR CUR.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /CRNT/ AIR(1000),AII(1000),BIR(1000),BII(1000),CIR(1000),CI
     1I(1000),CUR(2000)
      COMMON /JUNK/ NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
      COMMON /ANGL/ SALP(1000)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)
      COMPLEX CUR,CLO,CLL,CLY,AX,BX,CX
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      DATA TP/6.283185308/
      IF (N.EQ.0) GO TO 20
      DO 19 I=1,N
      CALL TRIO (I,JCO1,JCO2,DIL,DIK)
      CL=TP*DIL
      CK=TP*DIK
      SINL=SIN(CL)
      COSL=COS(CL)
      SINK=SIN(CK)
      COSK=COS(CK)
      SILK=SIN(CL+CK)
      CELLO=SINL+SINK-SILK
      IF (JCO1.GT.10000) GO TO 8
      IF (JCO1) 1,6,7
1     CLO=(0.,0.)
      IF (NCIX.LT.1) GO TO 3
      DO 2 K=1,NCIX
      JIXK=JIX(K)
2     CLO=CLO+CUR(JIXK)
3     CONTINUE
      IF (NCOX.LT.1) GO TO 5
      DO 4 K=1,NCOX
      JOXK=JOX(K)
4     CLO=CLO-CUR(JOXK)
5     CONTINUE
      GO TO 9
6     CLO=(0.,0.)
      GO TO 9
7     CLO=CUR(JCO1)
      IF (ICON2(JCO1).EQ.I) GO TO 9
      IF (JCO1.EQ.I) GO TO 9
      CLO=-CLO
      GO TO 9
8     CLO=CUR(I)
9     CLL=CUR(I)
      IF (JCO2.GT.10000) GO TO 17
      IF (JCO2) 10,15,16
10    CLY=(0.,0.)
      IF (NCOZ.LT.1) GO TO 12
      DO 11 K=1,NCOZ
      JOZK=JOZ(K)
11    CLY=CLY+CUR(JOZK)
12    CONTINUE
      IF (NCIZ.LT.1) GO TO 14
      DO 13 K=1,NCIZ
      JIZK=JIZ(K)
13    CLY=CLY-CUR(JIZK)
14    CONTINUE
      GO TO 18
15    CLY=(0.,0.)
      GO TO 18
16    CLY=CUR(JCO2)
      IF (ICON1(JCO2).EQ.I) GO TO 18
      IF (JCO2.EQ.I) GO TO 18
      CLY=-CLY
      GO TO 18
17    CLY=CUR(I)
```

```
18    AX=(CLO*SINK-CLL*SILK+CLY*SINL)/CELLO
      BX=(CLO*(COSK-1.)+CLL*(COSL-COSK)+CLY*(1.-COSL))/CELLO
      CX=-(CLO*SINK-CLL*(SINL+SINK)+CLY*SINL)/CELLO
      AIR(I)=REAL(AX)
      AII(I)=AIMAG(AX)
      BIR(I)=REAL(BX)
      BII(I)=AIMAG(BX)
      CIR(I)=REAL(CX)
19    CII(I)=AIMAG(CX)
20    IF (M.EQ.0) RETURN
C
C     SURFACE CURRENT ON EACH PATCH IS CONVERTED FROM TWO SURFACE
C     COMPONENTS TO THREE RECTANGULAR COMPONENTS
C
      K=LD-M
      JCO1=N+2*M+1
      JCO2=JCO1+M
      DO 21 I=1,M
      K=K+1
      JCO1=JCO1-2
      JCO2=JCO2-3
      CLO=CUR(JCO1)
      CLL=CUR(JCO1+1)
      CUR(JCO2)=CLO*T1X(K)+CLL*T2X(K)
      CUR(JCO2+1)=CLO*T1Y(K)+CLL*T2Y(K)
21    CUR(JCO2+2)=CLO*T1Z(K)+CLL*T2Z(K)
      RETURN
      END


      FUNCTION CANG (Z)
C
C     CANG RETURNS THE PHASE ANGLE OF A COMPLEX NUMBER IN DEGREES.
C
      COMPLEX Z
      CANG=ATGN2(AIMAG(Z),REAL(Z))*57.2957796
      RETURN
      END


      SUBROUTINE CATLOG
C
C     CATALOG CONTROLS THE WRITING OF FILES 11-16 ON FILE 17 WHEN THERE
C     IS A REQUEST TO INTERRUPT CALCULATION.  FILE 17 IS USED TO RESTART
C     THE PROGRAM.
C
      COMMON CM(4000)
      COMMON /SAVE/ IX(1500),IP(1500)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      DIMENSION IT(7)
      COMPLEX CM
      LOGICAL ENF
      EQUIVALENCE (N,NRES), (NP,NPRES), (I2,IBLCK)
      DATA (IT(I),I=1,7)/11,12,13,14,15,16,17/
      DO 1 I=1,7
      NUNIT=IT(I)
1     REWIND NUNIT
      WRITE (17) IC1,IC2,IC3,N,NP,I2,ICASE
      END FILE 17
      ISIX=6
      IF (ICASE.EQ.4) ISIX=1
      DO 4 I=1,ISIX
      NUNIT=IT(I)
      NEOF=777
      IFLCNT=0
2     CALL BLCKIN (NUNIT,1,I2,1,NEOF)
      IF (NEOF.EQ.0) GO TO 3
      CALL BLCKOT (17,1,I2,1,1)
      IFLCNT=IFLCNT+1
      GO TO 2
3     END FILE 17
      PRINT 10, IFLCNT,NUNIT
4     CONTINUE
      IF (ICASE.NE.4) GO TO 9
      IFLCNT=0
5     READ (13) (CM(I),I=1,NP)
      IF (ENF(13)) GO TO 6
      WRITE (17) (CM(I),I=1,NP)
      IFLCNT=IFLCNT+1
      GO TO 5
6     END FILE 17
      PRINT 10, IFLCNT,IT(3)
      IFLCNT=0
7     READ (15) (CM(I),I=1,NP)
      IF (ENF(15)) GO TO 8
      WRITE (17) (CM(I),I=1,NP)
      IFLCNT=IFLCNT+1
      GO TO 7
8     END FILE 17
      PRINT 10, IFLCNT,IT(5)
9     CONTINUE
      WRITE (17) (IX(I),I=1,N)
      WRITE (17) (IP(I),I=1,N)
      REWIND 17
      IF (IDUMP.EQ.0) STOP
      RETURN
```

```
C
10    FORMAT (/,I5,26H  FILES WRITTEN FROM UNIT ,I5/)
      END


      SUBROUTINE CHKPRT
C
C     CHKPRT CHECKS FOR INTERRUPT DURING OUT-OF-CORE MATRIX HANDLING.
C     THIS SUBROUTINE IS CALLED AT CONVENIENT PROGRAM INTERRUPT POINTS.
C
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      DATA ICK/0/
      PRINT 7, IC1,IC2,IC3
      IF (IDUMP.EQ.0) RETURN
C
C     AUTOMATIC FILE DUMPING
C
      CALL SECOND (T2)
      IF (ICK.EQ.0) T1=EXTIM
      ICK=1
      IF ((T2-T1).LT.TMDUM) RETURN
      T=T2-EXTIM
      PRINT 8, T
      CALL CATLOG
      PRINT 9
      T1=T2
C
C     REPOSITION FILES
C
      DO 1 I=11,16
1     BACKSPACE I
      IF (IC2.EQ.0) RETURN
      IF (ICASE-4) 2,3,5
2     RETURN
3     J=NRES-IC2*NPRES
      IF (J.EQ.0) RETURN
      DO 4 I=1,J
4     BACKSPACE 13
      RETURN
5     J=(NRES/NPRES-IC2)*NBLSYM
      IF (J.EQ.0) RETURN
      DO 6 I=1,J
6     BACKSPACE 11
      RETURN
C
7     FORMAT (12H CHECKPOINT ,4I5)
8     FORMAT (30H    FILE DUMP INITIATED, TIME=,F10.3,4H SEC)
9     FORMAT (22H    FILE DUMP COMPLETE)
      END


      SUBROUTINE CMSET (NROW,NCOL,CM,NLOAD,RKH)
C
C     CMSET SETS UP THE COMPLEX STRUCTURE MATRIX IN THE ARRAY CM
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION CM(NROW,NCOL)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      COMMON /ANGL/ SALP(1000)
      COMMON /JUNK/ NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
      COMMON /REFL/ RHOX,RHOY,RHOZ,CABJ,SABJ,SALPR,PX,PY,REFS,REFPS
      COMMON /ZLOAD/ ZARRAY(1000)
      COMMON /GND/ ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,IPERF
      DIMENSION CAB(1), SAB(1)
      DIMENSION ETR(3), ETI(3), TWHR(3,2), TWHI(3,2)
      EQUIVALENCE (ETR,TWHR), (ETI,TWHI), (CAB,ALP), (SAB,BET)
      COMPLEX FJ,CM,ZARRAY
      COMPLEX ZRATI,ZRATI2,REFS,REFPS,ZRSIN,ZRATIS,T1,ZSCRN
      DATA PI2/6.283185308/,ETA/376.73/,FJ/(0.,1.)/
      IF (ICASE.GT.2) REWIND 11
      I2=2*NPBLK*NROW
      IBLCK=I2
      IT=NPBLK
      NI1=IC1+1
      IF (IC1.EQ.0) GO TO 1
      IF (IC1.LE.(-1)) GO TO 25
      CALL BLCKIN (11,1,I2,IC1,1)
1     CONTINUE
C
C     CYCLE OVER MATRIX BLOCKS
C
      DO 24 IXBLK1=NI1,NBLOKS
      ISV=(IXBLK1-1)*NPBLK
      IF (IXBLK1.EQ.NBLOKS) IT=NLAST
      IF (ICASE.LT.3) IT=NCOL
      IF (NRADL.EQ.0) GO TO 2
      T1=FJ*2367.067/FLOAT(NRADL)
      T2=SCRWR*FLOAT(NRADL)
      ZRATIS=ZRATI
2     DO 3 I=1,NROW
      DO 3 J=1,IT
3     CM(I,J)=(0.,0.)
      IF (N.EQ.0) GO TO 20
```

```
C
C     WIRE SOURCE LOOP
C
      DO 19 J=1,N
      CALL TRIO (J,JCO1,JCO2,DIL,DIK)
      S=SI(J)
      B=BI(J)
      XJ=X(J)
      YJ=Y(J)
      ZJ=Z(J)
      CABJ=CAB(J)
      SABJ=SAB(J)
      SALPJ=SALP(J)
C
C     OBSERVATION LOOP
C
      DO 18 IPR=1,IT
      I=ISV+IPR
      IF (I.GT.NP) GO TO 14
C
C     WIRE OBSERVATION POINT
C
      XIJ=X(I)-XJ
      YIJ=Y(I)-YJ
      IJ=I-J
      CABI=CAB(I)
      SABI=SAB(I)
      SALPI=SALP(I)
      DO 4 IP=1,3
      ETR(IP)=0.
4     ETI(IP)=0.
      RFL=-1.
C
C     GROUND LOOP
C
      IFLG=0
      DO 13 IP=1,KSYMP
      RFL=-RFL
      SALPR=SALPJ*RFL
      ZIJ=Z(I)-RFL*ZJ
      ZP=XIJ*CABJ+YIJ*SABJ+ZIJ*SALPR
      DIJ=CABI*CABJ+SABI*SABJ+SALPI*SALPR
      RHOX=XIJ-CABJ*ZP
      RHOY=YIJ-SABJ*ZP
      RHOZ=ZIJ-SALPR*ZP
      RH=SQRT(RHOX*RHOX+RHOY*RHOY+RHOZ*RHOZ)
      IF (RH.GT.1.E-6) GO TO 5
      RHOX=0.
      RHOY=0.
      RHOZ=0.
      DIR=0.
      GO TO 6
5     RHOX=RHOX/RH
      RHOY=RHOY/RH
      RHOZ=RHOZ/RH
      DIR=RHOX*CABI+RHOY*SABI+RHOZ*SALPI
6     R=SQRT(XIJ*XIJ+YIJ*YIJ+ZIJ*ZIJ)
      IF (IP.NE.2) GO TO 11
      IJ=1
      IF (IPERF.EQ.1) GO TO 11
      RMAG=R
      XYMAG=SQRT(XIJ*XIJ+YIJ*YIJ)
C
C     SET PARAMETERS FOR RADIAL WIRE GROUND SCREEN.
C
      IF (NRADL.EQ.0) GO TO 8
      XSPEC=(X(I)*ZJ+Z(I)*XJ)/(Z(I)+ZJ)
      YSPEC=(Y(I)*ZJ+Z(I)*YJ)/(Z(I)+ZJ)
      RHOSPC=SQRT(XSPEC*XSPEC+YSPEC*YSPEC+T2*T2)
      IF (RHOSPC.GT.SCRWL) GO TO 7
      ZSCRN=T1*RHOSPC*ALOG(RHOSPC/T2)
      ZRATI=(ZSCRN*ZRATIS)/(ETA*ZRATIS+ZSCRN)
      GO TO 8
7     ZRATI=ZRATIS
8     IF (XYMAG.GT.1.E-6) GO TO 9
C
C     CALCULATION OF REFLECTION COEFFICIENTS WHEN GROUND IS SPECIFIED.
C
      PX=0.
      PY=0.
      CTH=1.
      ZRSIN=(1.,0.)
      GO TO 10
9     PX=-YIJ/XYMAG
      PY=XIJ/XYMAG
      CTH=ZIJ/RMAG
      ZRSIN=CSQRT(1.-ZRATI*ZRATI*(1.-CTH*CTH))
10    REFS=-(CTH-ZRATI*ZRSIN)/(CTH+ZRATI*ZRSIN)
      REFPS=(ZRATI*CTH-ZRSIN)/(ZRATI*CTH+ZRSIN)
      REFPS=REFPS-REFS
C
C     MATRIX ELEMENTS FOR WIRE SOURCE POINTS AND WIRE OBSERVATION POINTS
C
11    IF (R.GT.RKH) GO TO 12
      CALL INTG (B,S,RH,ZP,DIJ,DIR,ETR,ETI,DIL,DIK,IJ,IP)
      IFLG=1
      GO TO 13
12    CALL APRXE (R,S,ETA,PI2,DIJ,DIR,ZP,RH,ETR(2),ETI(2),IP)
13    CONTINUE
```

```
C
C     FILL MATRIX ELEMENTS.  ELEMENT LOCATIONS DETERMINED BY CONNECTION
C     DATA.
C
      CALL MATFIL (ETR,ETI,IPR,J,JCO1,JCO2,CM,NROW,NCOL,IFLG)
      GO TO 18
C     SURFACE OBSERVATION POINT
14    IK=I-NP
      IPATCH=(IK+1)/2
      IK=IK-(IK/2)*2
      IF (IK.EQ.0.AND.IPR.NE.1) GO TO 17
      DO 15 IP=1,3
      TWHR(IP,1)=0.
      TWHR(IP,2)=0.
      TWHI(IP,1)=0.
15    TWHI(IP,2)=0.
C
C     GROUND LOOP
C
      IFLG=0
      DO 16 IP=1,KSYMP
C
C     MATRIX ELEMENTS FOR WIRE SOURCE POINTS AND SURFACE OBSERVATION
C     POINTS
C
16    CALL HWMAT (XJ,YJ,ZJ,S,CABJ,SABJ,SALPJ,DIL,DIK,IPATCH,TWHR,TWHI,IP
     1,RKH,IFLG)
      IF (IK.EQ.0) GO TO 17
      CALL MATFIL (TWHR(1,1),TWHI(1,1),IPR,J,JCO1,JCO2,CM,NROW,NCOL,IFLG
     1)
      GO TO 18
17    CALL MATFIL (TWHR(1,2),TWHI(1,2),IPR,J,JCO1,JCO2,CM,NROW,NCOL,IFLG
     1)
18    CONTINUE
19    CONTINUE
      IF (NRADL.NE.0) ZRATI=ZRATIS
      IF (M.EQ.0) GO TO 21
20    IM1=ISV+1
      IM2=ISV+IT
C
C     ELEMENTS FOR SURFACE SOURCE POINTS AND WIRE OBSERVATION POINTS
C
      IF (IM1.LE.NP) CALL UDOTES (IM1,IM2,CM,NROW,NCOL)
C
C     ELEMENTS FOR SURFACE SOURCE AND OBSERVATION POINTS
C
      IF (IM2.GT.NP) CALL HMAT (IM1,IM2,CM,NROW,NCOL)
C
C     MATRIX ELEMENTS MODIFIED BY LOADING
C
21    IF (NLOAD.EQ.0) GO TO 23
      DO 22 I=1,IT
      J=ISV+I
      IF (J.GT.NP) GO TO 23
22    CM(J,I)=CM(J,I)-ZARRAY(J)
23    IF (ICASE.LT.3) GO TO 24
      CALL BLCKOT (11,1,12,1,31)
      IC1=IXBLK1
      IF (IC1.EQ.NBLOKS) GO TO 24
      CALL CHKPRT
24    CONTINUE
      IF (ICASE.LT.3) GO TO 25
      IC1=-2
      IF (ICASE.EQ.3) IC1=-1
      CALL CHKPRT
      REWIND 11
25    RETURN
      END


      SUBROUTINE CONECT (IGND)
C
C     CONNECT SETS UP SEGMENT CONNECTION DATA IN ARRAYS ICON1 AND ICON2
C     BY SEARCHING FOR SEGMENT ENDS THAT ARE IN CONTACT.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION X2(1), Y2(1), Z2(1)
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      SMIN=1.E-3
      JNO=0
      IF (IGND.EQ.0) GO TO 3
      PRINT 41
      IF (IGND.GT.0) PRINT 42
      IF (IPSYM.NE.2) GO TO 1
      NP=2*NP
      MP=2*MP
1     IF (IABS(IPSYM).LE.2) GO TO 2
      NP=N
      MP=M
2     IF (NP.GT.N) STOP
      IF (NP.EQ.N.AND.M.EQ.MP) IPSYM=0
3     IF (N.EQ.0) GO TO 36
      DO 32 I=1,N
      XI1=X(I)
      YI1=Y(I)
      ZI1=Z(I)
```

```
      XI2=X2(I)
      YI2=Y2(I)
      ZI2=Z2(I)
      SLEN=SQRT((XI2-XI1)**2+(YI2-YI1)**2+(ZI2-ZI1)**2)
C
C     DETERMINE CONNECTION DATA FOR END 1 OF SEGMENT.
C
      IF (IGND.LT.1) GO TO 5
      SEP=ZI1/SLEN
      IF (SEP.GT.(-SMIN)) GO TO 4
      PRINT 43, I
      STOP
4     IF (ABS(SEP).GT.SMIN) GO TO 5
      ICON1(I)=I
      GO TO 17
5     IF (ICON1(I).NE.0) GO TO 17
      DO 6 IC=1,N
      IF (IC.EQ.I) GO TO 6
      ISEG=IC
      SEP=(ABS(XI1-X(IC))+ABS(YI1-Y(IC))+ABS(ZI1-Z(IC)))/SLEN
      IF (SEP.LT.SMIN) GO TO 7
      SEP=(ABS(XI1-X2(IC))+ABS(YI1-Y2(IC))+ABS(ZI1-Z2(IC)))/SLEN
      IF (SEP.LT.SMIN) GO TO 12
6     CONTINUE
      GO TO 17
7     IF (ICON1(ISEG)) 9,8,10
8     ICON1(I)=ISEG
      ICON1(ISEG)=I
      GO TO 17
9     ICON1(I)=ICON1(ISEG)
      GO TO 17
10    JNO=JNO-1
      ICON1(I)=JNO
      IX=ICON1(ISEG)
      ICON1(ISEG)=JNO
      IF (ICON1(IX).EQ.ISEG) GO TO 11
      ICON2(IX)=JNO
      GO TO 17
11    ICON1(IX)=JNO
      GO TO 17
12    IF (ICON2(ISEG)) 14,13,15
13    ICON1(I)=ISEG
      ICON2(ISEG)=I
      GO TO 17
14    ICON1(I)=ICON2(ISEG)
      GO TO 17
15    JNO=JNO-1
      ICON1(I)=JNO
      IX=ICON2(ISEG)
      ICON2(ISEG)=JNO
      IF (ICON1(IX).EQ.ISEG) GO TO 16
      ICON2(IX)=JNO
      GO TO 17
16    ICON1(IX)=JNO
C
C     DETERMINE CONNECTION DATA FOR END 2 OF SEGMENT.
C
17    IF (IGND.LT.1) GO TO 20
      SEP=ZI2/SLEN
      IF (SEP.GT.(-SMIN)) GO TO 18
      PRINT 43, I
      STOP
18    IF (ABS(SEP).GT.SMIN) GO TO 20
      IF (ICON1(I).NE.I) GO TO 19
      PRINT 44, I
      STOP
19    ICON2(I)=I
      GO TO 32
20    IF (ICON2(I).NE.0) GO TO 32
      DO 21 IC=1,N
      IF (IC.EQ.I) GO TO 21
      ISEG=IC
      SEP=(ABS(XI2-X(IC))+ABS(YI2-Y(IC))+ABS(ZI2-Z(IC)))/SLEN
      IF (SEP.LT.SMIN) GO TO 22
      SEP=(ABS(XI2-X2(IC))+ABS(YI2-Y2(IC))+ABS(ZI2-Z2(IC)))/SLEN
      IF (SEP.LT.SMIN) GO TO 27
21    CONTINUE
      GO TO 32
22    IF (ICON1(ISEG)) 24,23,25
23    ICON2(I)=ISEG
      ICON1(ISEG)=I
      GO TO 32
24    ICON2(I)=ICON1(ISEG)
      GO TO 32
25    JNO=JNO-1
      ICON2(I)=JNO
      IX=ICON1(ISEG)
      ICON1(ISEG)=JNO
      IF (ICON1(IX).EQ.ISEG) GO TO 26
      ICON2(IX)=JNO
      GO TO 32
26    ICON1(IX)=JNO
      GO TO 32
27    IF (ICON2(ISEG)) 29,28,30
28    ICON2(I)=ISEG
      ICON2(ISEG)=I
      GO TO 32
29    ICON2(I)=ICON2(ISEG)
      GO TO 32
```

```
30    JNO=JNO-1
      ICON2(I)=JNO
      IX=ICON2(ISEG)
      ICON2(ISEG)=JNO
      IF (ICON1(IX).EQ.ISEG) GO TO 31
      ICON2(IX)=JNO
      GO TO 32
31    ICON1(IX)=JNO
32    CONTINUE
      IF (M.EQ.0) GO TO 36
C
C     SEARCH FOR SEGMENT ENDS CONNECTED TO PATCH CENTERS.  WHEN FOUND
C     SET CONNECTION DATA FOR SEGMENT END AND SUBDIVIDE PATCH.
C
      SSMIN=SMIN*SMIN
      IX=LD+1
      DO 35 I=1,M
      IX=IX-1
      XS=X(IX)
      YS=Y(IX)
      ZS=Z(IX)
      DO 34 ISEG=1,N
      XI1=X(ISEG)
      YI1=Y(ISEG)
      ZI1=Z(ISEG)
      XI2=X2(ISEG)
      YI2=Y2(ISEG)
      ZI2=Z2(ISEG)
      SLEN=((XI2-XI1)**2+(YI2-YI1)**2+(ZI2-ZI1)**2)*SSMIN
      SEP=(XI1-XS)**2+(YI1-YS)**2+(ZI1-ZS)**2
      IF (SEP.GT.SLEN) GO TO 33
      ICON1(ISEG)=10000+I
      CALL SUBPH (I,IC,XI1,YI1,ZI1,XI2,YI2,ZI2)
      GO TO 35
33    SEP=(XI2-XS)**2+(YI2-YS)**2+(ZI2-ZS)**2
      IF (SEP.GT.SLEN) GO TO 34
      ICON2(ISEG)=10000+I
      CALL SUBPH (I,IC,XI1,YI1,ZI1,XI2,YI2,ZI2)
      GO TO 35
34    CONTINUE
35    CONTINUE
36    PRINT 45, N,NP,IPSYM
      IF (M.GT.0) PRINT 48, M,MP
      IF (N.EQ.NP) GO TO 40
      ISEG=N/NP
      IF (IPSYM) 38,37,39
37    STOP
38    PRINT 46, ISEG
      GO TO 40
39    IC=ISEG/2
      IF (ISEG.EQ.8) IC=3
      PRINT 47, IC
40    CONTINUE
      RETURN
C
41    FORMAT (/,3X,23HGROUND PLANE SPECIFIED.)
42    FORMAT (/,3X,46HWHERE WIRE ENDS TOUCH GROUND, CURRENT WILL BE ,38H
     1INTERPOLATED TO IMAGE IN GROUND PLANE.,/)
43    FORMAT (30H GEOMETRY DATA ERROR-- SEGMENT,I5,21H EXTENDS BELOW GRO
     1UND)
44    FORMAT (29H GEOMETRY DATA ERROR--SEGMENT,I5,16H LIES IN GROUND ,6H
     1PLANE.)
45    FORMAT (/,3X,20HTOTAL SEGMENTS USED=,I5,5X,12HNO. SEG. IN ,17HA SY
     1MMETRIC CELL=,I5,5X,14HSYMMETRY FLAG=,I3)
46    FORMAT (14H STRUCTURE HAS,I4,25H FOLD ROTATIONAL SYMMETRY,/)
47    FORMAT (14H STRUCTURE HAS,I2,19H PLANES OF SYMMETRY,/)
48    FORMAT (3X,19HTOTAL PATCHES USED=,I5,6X,32HNO. PATCHES IN A SYMMET
     1RIC CELL=,I5)
      END
```

```
      SUBROUTINE CONVRT
C
C     CONVERT CHANGES GEOMETRY DATA FROM THE FORM STATING X,Y,Z
C     COORDINATES OF EACH SEGMENT END TO X,Y,Z OF THE SEGMENT CENTER
C     PLUS SEGMENT LENGTH AND ORIENTATION ANGLES AS REQUIRED IN MAIN
C     PROGRAM.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION X2(1), Y2(1), Z2(1)
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      IF (N.EQ.0) RETURN
      DO 1 I=1,N
      XA=X(I)
      YA=Y(I)
      ZA=Z(I)
      XB=X2(I)
      YB=Y2(I)
      ZB=Z2(I)
      X(I)=(XA+XB)*.5
      Y(I)=(YA+YB)*.5
      Z(I)=(ZA+ZB)*.5
      XA=XB-XA
      YA=YB-YA
      ZA=ZB-ZA
      SI(I)=SQRT(XA*XA+YA*YA+ZA*ZA)
```

```
      ALP(I)=ASIN(ZA/SI(I))
1     BET(I)=ATGN2(YA,XA)
      RETURN
      END


      SUBROUTINE DATAGN
C
C     DATAGN IS THE MAIN ROUTINE FOR INPUT OF GEOMETRY DATA.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION X2(1), Y2(1), Z2(1)
      DIMENSION ATST(9), IFX(2), IFY(2), IFZ(2)
      INTEGER GM,ATST
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      DATA ATST/2HGW,2HGX,2HGR,2HGS,2HGE,2HGM,2HSP,2HSS,2HGC/
      DATA IFX/1H ,1HX/,IFY/1H ,1HY/,IFZ/1H ,1HZ/
      DATA TA/0.01745329252/
      IPSYM=0
      NWIRE=0
      N=0
      NP=0
      M=0
      MP=0
      DO 1 I=1,LD
      ICON1(I)=0
1     ICON2(I)=0
      PRINT 21
      PRINT 22
C
C     READ GEOMETRY DATA CARD AND BRANCH TO SECTION FOR OPERATION
C     REQUESTED
C
2     READ 23, GM,ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD
      IF (N+M.GT.LD) GO TO 20
      IF (GM.EQ.ATST(1)) GO TO 3
      IF (GM.EQ.ATST(2)) GO TO 6
      IF (GM.EQ.ATST(3)) GO TO 7
      IF (GM.EQ.ATST(4)) GO TO 9
      IF (GM.EQ.ATST(7)) GO TO 4
      IF (GM.EQ.ATST(8)) GO TO 5
      IF (GM.EQ.ATST(5)) GO TO 17
      IF (GM.EQ.ATST(6)) GO TO 14
      IF (GM.EQ.ATST(9)) GO TO 15
      GO TO 19
C
C     GENERATE SEGMENT DATA FOR STRAIGHT WIRE.
C
3     NWIRE=NWIRE+1
      I1=N+1
      I2=N+NS
      PRINT 24, NWIRE,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,NS,I1,I2,ITG
      CALL WIRE (XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,NS,ITG)
      GO TO 2
C
C     GENERATE DATA FOR A SURFACE PATCH
C
4     I1=M+1
      PRINT 32, I1,XW1,YW1,ZW1,XW2,YW2,ZW2
      XW2=XW2*TA
      YW2=YW2*TA
      CALL PATCH (ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2)
      GO TO 2
C
C     FORM SURFACE BY MULTIPLE SHIFTING OF LAST PATCH INPUT.
C
5     PRINT 33, ITG,XW1,NS,YW1
      CALL PACHS (ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2)
      GO TO 2
C
C     REFLECT STRUCTURE ALONG X,Y, OR Z AXES OR ROTATE TO FORM CYLINDER.
C
6     IY=NS/10
      IZ=NS-IY*10
      IX=IY/10
      IY=IY-IX*10
      IF (IX.NE.0) IX=1
      IF (IY.NE.0) IY=1
      IF (IZ.NE.0) IZ=1
      PRINT 25, IFX(IX+1),IFY(IY+1),IFZ(IZ+1),ITG
      GO TO 8
7     PRINT 26, NS,ITG
      IX=-1
8     CALL REFLC (IX,IY,IZ,ITG,NS)
      GO TO 2
C
C     SCALE STRUCTURE DIMENSIONS BY FACTOR XW1.
C
9     IF (N.EQ.0) GO TO 11
      DO 10 I=1,N
      X(I)=X(I)*XW1
      Y(I)=Y(I)*XW1
      Z(I)=Z(I)*XW1
      X2(I)=X2(I)*XW1
      Y2(I)=Y2(I)*XW1
      Z2(I)=Z2(I)*XW1
```

```
10    BI(I)=BI(I)*XW1
11    IF (M.EQ.0) GO TO 13
      YW1=XW1*XW1
      IX=LD+1-M
      DO 12 I=IX,LD
      X(I)=X(I)*XW1
      Y(I)=Y(I)*XW1
      Z(I)=Z(I)*XW1
12    BI(I)=BI(I)*YW1
13    PRINT 27, XW1
      GO TO 2
C
C     MOVE STRUCTURE OR REPRODUCE ORIGINAL STRUCTURE IN NEW POSITIONS.
C
14    PRINT 28, ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD
      XW1=XW1*TA
      YW1=YW1*TA
      ZW1=ZW1*TA
      CALL MOVE (XW1,YW1,ZW1,XW2,YW2,ZW2,INT(RAD+.5),NS,ITG)
      GO TO 2
15    I=ISEGNO(ITG,NS)
      IX=XW1+.5
      IF (IX.EQ.2) GO TO 16
      ICON1(I)=I
      GO TO 2
16    ICON2(I)=I
      GO TO 2
C
C     TERMINATE STRUCTURE GEOMETRY INPUT.
C
17    IF (N.EQ.0) GO TO 18
      CALL CONECT (ITG)
      CALL CONVRT
18    IF (N+M.GT.LD) GO TO 20
      RETURN
19    PRINT 29
      PRINT 30, GM,ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD
      STOP
20    PRINT 31
      STOP
C
21    FORMAT (////,33X,35H- - - STRUCTURE SPECIFICATION - - -,//,37X,28H
     1COORDINATES MUST BE INPUT IN,/,37X,29HMETERS OR BE SCALED TO METER
     2S,/,37X,31HBEFORE STRUCTURE INPUT IS ENDED,//)
22    FORMAT (2X,4HWIRE,79X,6HNO. OF,4X,5HFIRST,2X,4HLAST,5X,3HTAG,/,2X,
     13HNO.,8X,2HX1,9X,2HY1,9X,2HZ1,10X,2HX2,9X,2HY2,9X,2HZ2,6X,6HRADIUS
     2,3X,4HSEG.,5X,4HSEG.,3X,4HSEG.,5X,3HNO.)
23    FORMAT (A2,I3,I5,7F10.5)
24    FORMAT (1X,I5,3F11.5,1X,4F11.5,2X,I5,4X,I5,1X,I5,3X,I5)
25    FORMAT (6X,34HSTRUCTURE REFLECTED ALONG THE AXES,3(1X,A1),22H.  TA
     1GS INCREMENTED BY,I5)
26    FORMAT (6X,30HSTRUCTURE ROTATED ABOUT Z-AXIS,I3,30H TIMES.  LABLES
     1 INCREMENTED BY,I5)
27    FORMAT (6X,26HSTRUCTURE SCALED BY FACTOR,F10.5)
28    FORMAT (6X,49HTHE STRUCTURE HAS BEEN MOVED, MOVE DATA CARD IS -/6X
     1,I3,I5,7F10.5)
29    FORMAT (25H GEOMETRY DATA CARD ERROR)
30    FORMAT (1X,A2,I3,I5,7F10.5)
31    FORMAT (69H NUMBER OF WIRE SEGMENTS AND SURFACE PATCHES EXCEEDS DI
     1MENSION LIMIT.)
32    FORMAT (1X,I5,1HP,F10.5,2F11.5,1X,2F11.5,11X,F11.5)
33    FORMAT (6X,29HLAST SURFACE PATCH REPRODUCED,I3,36H TIMES IN X DIRE
     1CTION WITH INCREMENT,F11.6,/,32X,3HAND,I3,36H TIMES IN Y DIRECTION
     2 WITH INCREMENT,F11.6)
      END
```

```
      FUNCTION DB10 (X)
C
C     FUNCTION DB-- RETURNS DB FOR MAGNITUDE (FIELD) OR MAG**2 (POWER) I
C
      F=10.
      GO TO 1
      ENTRY DB20
      F=20.
1     IF (X.LT.1.E-20) GO TO 2
      DB10=F*ALOG10(X)
      RETURN
2     DB10=-999.99
      RETURN
      END
```

```
      SUBROUTINE EFLD (B,S,RH,ZP,IJ,EZRS,EZIS,ERRS,ERIS,EZRC,EZIC,ERRC,E
     1RIC,EZRK,EZIK,ERRK,ERIK)
C
C     EFLD RETURNS ELECTRIC FIELD OF A SEGMENT IN RHO AND Z COMPONENTS
C     (CYLN. COORD. CENTERED ON SOURCE SEGMENT) FOR CONSTANT, SINE, AND
C     COSINE CURRENT DISTRIBUTIONS ON THE SEGMENT.
C
      COMMON /TMI/ ZPK,RKB2,IJX
      DATA ZZ,TP/188.363635,6.283185308/
      IJX=IJ
      RHK=RH*TP
      ZPK=ZP*TP
      BK=B*TP
      RKB2=RHK*RHK+BK*BK
```

```
      RKB=SQRT(RKB2)
      COINC=RHK/RKB
      SKT=TP*S*0.5
      ZD2=ZPK-SKT
      ZD1=ZPK+SKT
      R2KS=RKB2+ZD2*ZD2
      R2K=SQRT(R2KS)
      R1KS=RKB2+ZD1*ZD1
      R1K=SQRT(R1KS)
      SR2=SIN(R2K)/R2K*ZZ
      CR2=COS(R2K)/R2K*ZZ
      SR1=SIN(R1K)/R1K*ZZ
      CR1=COS(R1K)/R1K*ZZ
      SR2R=SR2/R2K
      SR2RR=SR2/R2KS
      CR2R=CR2/R2K
      CR2RR=CR2/R2KS
      SR1R=SR1/R1K
      SR1RR=SR1/R1KS
      CR1R=CR1/R1K
      CR1RR=CR1/R1KS
      CST=COS(SKT)
      SST=SIN(SKT)
      T1=(CR2R-SR2RR)*ZD2
      T2=(CR1R-SR1RR)*ZD1
      T3=(SR2R+CR2RR)*ZD2
      T4=(SR1R+CR1RR)*ZD1
      T1S=T1*SST
      T2S=-T2*SST
      T3S=T3*SST
      T4S=-T4*SST
      EZRS=(SR2-SR1)*CST+T1S-T2S
      EZIS=(CR2-CR1)*CST-T3S+T4S
      ERRS=-((SR2*ZD2-SR1*ZD1)*CST+(SR2+SR1)*SST+T1S*ZD2-T2S*ZD1)/RKB*CO
     1INC
      ERIS=-((CR2*ZD2-CR1*ZD1)*CST+(CR2+CR1)*SST-T3S*ZD2+T4S*ZD1)/RKB*CO
     1INC
      T1S=T1*CST
      T2S=T2*CST
      T3S=T3*CST
      T4S=T4*CST
      EZRC=(-(SR2+SR1)*SST+T1S-T2S)
      EZIC=(-(CR2+CR1)*SST-T3S+T4S)
      ERRC=-(-(SR2*ZD2+SR1*ZD1)*SST+(SR2-SR1)*CST+T1S*ZD2-T2S*ZD1)/RKB*C
     1OINC
      ERIC=-(-(CR2*ZD2+CR1*ZD1)*SST+(CR2-CR1)*CST-T3S*ZD2+T4S*ZD1)/RKB*C
     1OINC
      ERRK=RKB*(CR2R-SR2RR-CR1R+SR1RR)*COINC
      ERIK=-RKB*(SR2R+CR2RR-SR1R-CR1RR)*COINC
      CALL INTX (-SKT,SKT,BK,IJ,CINT,SINT)
      EZRK=-ZZ*SINT+T1-T2
      EZIK=-ZZ*CINT-T3+T4
      RETURN
      END


      COMPLEX FUNCTION EFUN(ZX)
C
C     EFUN(Z)=SQRT(PI)*Z*EXP(Z*Z)*(1.-ERF(Z)).  ERF(Z) IS THE ERROR
C     FUNCTION.
C
      COMPLEX Z,ZS,SUM,POW,TERM,ZX
      DATA TOSP/1.12837917/,ACCS/1.E-16/,SP/1.77245385/
      Z=ZX
      IF (CABS(ZX).GT.3.) GO TO 3
C
C     SERIES EXPANSION
C
      ZS=Z*Z
      SUM=Z
      POW=Z
      DO 1 I=1,100
      FI=1./I
      POW=-POW*ZS*FI
      FI=1./(2.*I+1.)
      TERM=POW*FI
      SUM=SUM+TERM
      TMS=REAL(TERM*CONJG(TERM))
      SMS=REAL(SUM*CONJG(SUM))
      IF (TMS/SMS.LT.ACCS) GO TO 2
1     CONTINUE
      PRINT 8, ZX
2     EFUN=(1.-SUM*TOSP)*Z*CEXP(ZS)*SP
      RETURN
C
C     ASYMPTOTIC EXPANSION
C
3     IF (REAL(Z).GE.0.) GO TO 4
      MINUS=1
      Z=-ZX
      GO TO 5
4     MINUS=0
5     ZS=.5/(Z*Z)
      RATL=1.E+5
      SUM=(1.,0.)
      TERM=(1.,0.)
      DO 6 I=1,100
      TERM=-TERM*(2.*I-1.)*ZS
```

```
      TMS=REAL(TERM*CONJG(TERM))
      SMS=REAL(SUM*CONJG(SUM))
      RAT=TMS/SMS
      IF (RAT.GT.RATL) GO TO 7
      SUM=SUM+TERM
      RATL=RAT
      IF (RAT.LT.ACCS) GO TO 7
6     CONTINUE
      PRINT 8, ZX
7     IF (MINUS.EQ.1) SUM=SUM-2.*SP*Z*CEXP(Z*Z)
      EFUN=SUM
      RETURN
C
8     FORMAT (38H SERIES DID NOT CONVERGE IN EFUN, ARG=,2E13.5)
      END
```

```
      LOGICAL FUNCTION ENF(NUNIT)
C
C     FUNCTION ENF CHECKS FOR END OF FILE
C
      IF(EOF(NUNIT)) 2,1
1     ENF=.FALSE.
      RETURN
2     ENF=.TRUE.
      RETURN
      END
```

```
      SUBROUTINE ETMNS (P1,P2,P3,P4,P5,P6,IPR,ISANT,VSANT,NSANT,E)
C
C     ETMNS FILLS THE ARRAY E WITH THE NEGATIVE OF THE ELECTRIC FIELD
C     INCIDENT ON WIRES AND THE TANGENTIAL MAGNETIC FIELD  (N X H) ON
C     SURFACES.  E IS THE RIGHT HAND SIDE OF THE MATRIX EQUATION.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      DIMENSION CAB(1), SAB(1)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)
      DIMENSION E(1500), VSANT(10), ISANT(10)
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1))
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      COMPLEX E,CX,CY,CZ,VSANT,T1,T2,ER,ET,EZH,ERH
      DATA TP/6.2831853/,RETA/0.0026544/
      NEQ=N+M*2
      IF (IPR.GT.0) GO TO 3
C
C     APPLIED FIELD OF VOLTAGE SOURCES FOR TRANSMITTING CASE
C
      DO 1 I=1,NEQ
1     E(I)=(0.,0.)
      DO 2 I=1,NSANT
      IS=ISANT(I)
2     E(IS)=-VSANT(I)/(SI(IS)*WLAM)
      RETURN
3     IF (IPR.GT.3) GO TO 11
C
C     INCIDENT PLANE WAVE, LINEARLY POLARIZED.
C     ELECTRIC FIELD ON WIRES
C
      CTH=COS(P1)
      STH=SIN(P1)
      CPH=COS(P2)
      SPH=SIN(P2)
      CET=COS(P3)
      SET=SIN(P3)
      PX=CTH*CPH*CET-SPH*SET
      PY=CTH*SPH*CET+CPH*SET
      PZ=-STH*CET
      WX=-STH*CPH
      WY=-STH*SPH
      WZ=-CTH
      QX=WY*PZ-WZ*PY
      QY=WZ*PX-WX*PZ
      QZ=WX*PY-WY*PX
      IF (IPR.GT.1) GO TO 7
      IF (N.EQ.0) GO TO 5
      DO 4 I=1,N
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
4     E(I)=-(PX*CAB(I)+PY*SAB(I)+PZ*SALP(I))*CMPLX(COS(ARG),SIN(ARG))
5     IF (M.EQ.0) RETURN
C     MAGNETIC FIELD ON SURFACES
      I=LD+1
      I1=N-1
      DO 6 IS=1,M
      I=I-1
      I1=I1+2
      I2=I1+1
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
      T1=CMPLX(COS(ARG),SIN(ARG))*SALP(I)*RETA
      E(I2)=(QX*T1X(I)+QY*T1Y(I)+QZ*T1Z(I))*T1
6     E(I1)=(QX*T2X(I)+QY*T2Y(I)+QZ*T2Z(I))*T1
      RETURN
C
```

```
C     INCIDENT PLANE WAVE, ELLIPTIC POLARIZATION.
C     ELECTRIC FIELD ON WIRES
C
7     T1=-(0.,1.)*P6
      IF (IPR.EQ.3) T1=-T1
      IF (N.EQ.0) GO TO 9
      CX=PX+T1*QX
      CY=PY+T1*QY
      CZ=PZ+T1*QZ
      DO 8 I=1,N
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
8     E(I)=-(CX*CAB(I)+CY*SAB(I)+CZ*SALP(I))*CMPLX(COS(ARG),SIN(ARG))
9     IF (M.EQ.0) RETURN
C     MAGNETIC FIELD ON SURFACES
      CX=QX-T1*PX
      CY=QY-T1*PY
      CZ=QZ-T1*PZ
      I=LD+1
      I1=N-1
      DO 10 IS=1,M
      I=I-1
      I1=I1+2
      I2=I1+1
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
      T2=CMPLX(COS(ARG),SIN(ARG))*SALP(I)*RETA
      E(I2)=(CX*T1X(I)+CY*T1Y(I)+CZ*T1Z(I))*T2
10    E(I1)=(CX*T2X(I)+CY*T2Y(I)+CZ*T2Z(I))*T2
      RETURN
C
C     INCIDENT FIELD OF AN ELEMENTARY CURRENT SOURCE.
C
11    WZ=COS(P4)
      WX=WZ*COS(P5)
      WY=WZ*SIN(P5)
      WZ=SIN(P4)
      DS=P6*59.958
      DSH=P6/(2.*TP)
      NPM=N+M
      IS=LD+1
      I1=N-1
      DO 16 I=1,NPM
      II=I
      IF (I.LE.N) GO TO 12
      IS=IS-1
      II=IS
      I1=I1+2
      I2=I1+1
12    PX=X(II)-P1
      PY=Y(II)-P2
      PZ=Z(II)-P3
      RS=PX*PX+PY*PY+PZ*PZ
      IF (RS.LT.1.E-30) GO TO 16
      R=SQRT(RS)
      PX=PX/R
      PY=PY/R
      PZ=PZ/R
      CTH=PX*WX+PY*WY+PZ*WZ
      STH=SQRT(1.-CTH*CTH)
      QX=PX-WX*CTH
      QY=PY-WY*CTH
      QZ=PZ-WZ*CTH
      ARG=SQRT(QX*QX+QY*QY+QZ*QZ)
      IF (ARG.LT.1.E-30) GO TO 13
      QX=QX/ARG
      QY=QY/ARG
      QZ=QZ/ARG
      GO TO 14
13    QX=1.
      QY=0.
      QZ=0.
14    ARG=-TP*R
      T1=CMPLX(COS(ARG),SIN(ARG))
      IF (I.GT.N) GO TO 15
C     ELECTRIC FIELD ON WIRES
      T2=CMPLX(1.,-1./(R*TP))/RS
      ER=DS*T1*T2*CTH
      ET=.5*DS*T1*((0.,1.)*TP/R+T2)*STH
      EZH=ER*CTH-ET*STH
      ERH=ER*STH+ET*CTH
      CX=EZH*WX+ERH*QX
      CY=EZH*WY+ERH*QY
      CZ=EZH*WZ+ERH*QZ
      E(I)=-(CX*CAB(I)+CY*SAB(I)+CZ*SALP(I))
      GO TO 16
C     MAGNETIC FIELD ON SURFACES
15    PX=WY*QZ-WZ*QY
      PY=WZ*QX-WX*QZ
      PZ=WX*QY-WY*QX
      T2=DSH*T1*CMPLX(1./R,TP)/R*STH*SALP(II)
      CX=T2*PX
      CY=T2*PY
      CZ=T2*PZ
      E(I2)=CX*T1X(II)+CY*T1Y(II)+CZ*T1Z(II)
      E(I1)=CX*T2X(II)+CY*T2Y(II)+CZ*T2Z(II)
16    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FACIO (A,NROW,NCOL,IP)
C
C     FACIO CONTROLS I/O FOR OUT-OF-CORE FACTORIZATION.
C
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      COMPLEX A
      DIMENSION A(NROW,NCOL), IP(NROW)
      IT=2*NPBLK*NROW
      I1=1
      I2=IT
      I3=I2+1
      I4=2*IT
      REWIND 12
      REWIND 13
      REWIND 14
      TIME=0.
      IF (IC3.EQ.-1) GO TO 4
      IF (IC3.EQ.0) GO TO 1
      CALL BLCKIN (11,1,I2,NBLOKS,998)
      CALL BLCKIN (12,1,I2,IC3,2)
      IXBLK1=IC3
      GO TO 5
    1 CONTINUE
C
C     BUFFER IN BLOCK1 AND BLOCK2 FROM TAPE 1
C
      CALL BLCKIN (11,I1,I2,1,17)
      CALL BLCKIN (11,I3,I4,1,18)
      CALL SECOND (T1)
      CALL LFACTR (A,NROW,NCOL,1,2,IP)
      CALL SECOND (T2)
      TIME=TIME+(T2-T1)
C
C     BUFFER OUT BLOCK1 TO TAPE2 (BLOCK1 FACTORED)
C
      CALL BLCKOT (12,I1,I2,1,3)
      IF (NBLOKS.LT.3) GO TO 3
C
C     BUFFER OUT BLOCK2 TO FILE3
C
      CALL BLCKOT (13,I3,I4,1,20)
      DO 2 IXBLK2=3,NBLOKS
C
C     BUFFER IN BLOCK2 FROM TAPE1
C
      CALL BLCKIN (11,I3,I4,1,21)
      CALL SECOND (T1)
      CALL LFACTR (A,NROW,NCOL,1,IXBLK2,IP)
      CALL SECOND (T2)
      TIME=TIME+(T2-T1)
C
C     BUFFER OUT BLOCK2 TO FILE3
C
      CALL BLCKOT (13,I3,I4,1,22)
    2 CONTINUE
      GO TO 4
    3 CONTINUE
      CALL BLCKOT (12,I3,I4,1,23)
      PRINT 10, TIME
      REWIND 12
      REWIND 13
      REWIND 14
      GO TO 7
    4 IXBLK1=1
    5 IXBLK1=IXBLK1+1
      IC3=IXBLK1-1
      CALL CHKPRT
      IXBLK2=IXBLK1+1
C
C     WITH THE EXCEPTION OF THE FIRST PASS, IFILE3 BECOMES IFILE4 AND
C     VISA VERSA.
C
      IFILE3=14
      IFILE4=13
      IF (2*(IXBLK1/2).NE.IXBLK1) GO TO 6
      IFILE3=13
      IFILE4=14
    6 REWIND IFILE3
      REWIND IFILE4
C
C     BUFFER IN BLOCK1 AND BLOCK2 FROM IFILE3
C
      CALL BLCKIN (IFILE3,I1,I2,1,24)
      CALL BLCKIN (IFILE3,I3,I4,1,25)
      CALL SECOND (T1)
      CALL LFACTR (A,NROW,NCOL,IXBLK1,IXBLK2,IP)
      CALL SECOND (T2)
      TIME=TIME+(T2-T1)
C
C     BUFFER OUT BLOCK1 TO TAPE2 (BLOCK1 FACTORED)
C
      CALL BLCKOT (12,I1,I2,1,26)
C
C     BUFFER OUT BLOCK2 TO IFILE4
C
      CALL BLCKOT (IFILE4,I3,I4,1,27)
      IF (IXBLK2.NE.NBLOKS) GO TO 9
```

```
C
C     BUFFER OUT BLOCK2 TO TAPE2 (BLOCK1 FACTORED--FACTORIZATION
C     FINISHED)
C
      CALL BLCKOT (12,13,14,1,28)
      PRINT 10, TIME
      REWIND 12
      REWIND 13
      REWIND 14
7     CONTINUE
      IC3=0
8     RETURN
9     IXBLK2=IXBLK2+1
      IF (IXBLK2.GT.NBLOKS) GO TO 5
C
C     BUFFER IN BLOCK2 FROM IFILE3
C
      CALL BLCKIN (IFILE3,I3,I4,1,29)
      CALL SECOND (T1)
      CALL LFACTR (A,NROW,NCOL,IXBLK1,IXBLK2,IP)
      CALL SECOND (T2)
      TIME=TIME+(T2-T1)
C
C     BUFFER OUT BLOCK2 TO FILE4
C
      CALL BLCKOT (IFILE4,I3,I4,1,30)
      GO TO 9
C
10    FORMAT (35H CP TIME TAKEN FOR FACTORIZATION = ,E12.5)
      END


      SUBROUTINE FACTR (N,A,IP,NDIM)
C
C     SUBROUTINE TO FACTOR A MATRIX INTO A UNIT LOWER TRIANGULAR MATRIX
C     AND AN UPPER TRIANGULAR MATRIX USING THE GAUSS-DOOLITTLE ALGORITHM
C     PRESENTED ON PAGES 411-416 OF A. RALSTON--A FIRST COURSE IN
C     NUMERICAL ANALYSIS.  COMMENTS BELOW REFER TO COMMENTS IN RALSTONS
C     TEXT.     (MATRIX TRANSPOSED.
C
      DIMENSION A(NDIM,NDIM), IP(NDIM)
      COMMON /SCRATM/ D(1500)
      COMPLEX A,D,ARJ
      INTEGER R,RM1,RP1,PJ,PR
      IFLG=0
      DO 9 R=1,N
C
C     STEP 1
C
      DO 1 K=1,N
      D(K)=A(R,K)
1     CONTINUE
C
C     STEPS 2 AND 3
C
      RM1=R-1
      IF (RM1.LT.1) GO TO 4
      DO 3 J=1,RM1
      PJ=IP(J)
      ARJ=D(PJ)
      A(R,J)=ARJ
      D(PJ)=D(J)
      JP1=J+1
      DO 2 I=JP1,N
      D(I)=D(I)-A(J,I)*ARJ
2     CONTINUE
3     CONTINUE
4     CONTINUE
C
C     STEP 4
C
      DMAX=REAL(D(R)*CONJG(D(R)))
      IP(R)=R
      RP1=R+1
      IF (RP1.GT.N) GO TO 6
      DO 5 I=RP1,N
      ELMAG=REAL(D(I)*CONJG(D(I)))
      IF (ELMAG.LT.DMAX) GO TO 5
      DMAX=ELMAG
      IP(R)=I
5     CONTINUE
6     CONTINUE
      IF (DMAX.LT.1.E-10) IFLG=1
      PR=IP(R)
      A(R,R)=D(PR)
      D(PR)=D(R)
C
C     STEP 5
C
      IF (RP1.GT.N) GO TO 8
      ARJ=1./A(R,R)
      DO 7 I=RP1,N
      A(R,I)=D(I)*ARJ
7     CONTINUE
8     CONTINUE
      IF (IFLG.EQ.0) GO TO 9
      PRINT 10, R,DMAX
      IFLG=0
```

```
9     CONTINUE
      RETURN
C
10    FORMAT (1H ,6HPIVOT(,I3,2H)=,E16.8)
      END


      SUBROUTINE FACTRS (NP,NOP,A,IP,IX,NROW,NCOL,NCOLS,IPSYM)
C
C     FACTRS, FOR SYMMETRIC STRUCTURE, TRANSFORMS SUBMATRICIES TO FORM
C     MATRICIES OF THE SYMMETRIC MODES AND CALLS ROUTINE TO FACTOR
C     MATRICIES.  IF NO SYMMETRY, THE ROUTINE IS CALLED TO FACTOR THE
C     COMPLETE MATRIX.
C
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /SMAT/ S(10,10)
      COMMON /SCRATM/ D(1500)
      DIMENSION A(NROW,NCOL), IP(NROW), IX(NROW)
      COMPLEX A,D,DETER,S
      IF (ICASE.LT.3) GO TO 1
      REWIND 12
      REWIND 13
      REWIND 14
      REWIND 15
      REWIND 16
1     IF (NOP.EQ.1) GO TO 14
      IF (IPSYM.GT.0) GO TO 3
C
C     SET UP S MATRIX FOR ROTATIONAL SYMMETRY.
C
      PHAZ=6.2831853072/NOP
      DO 2 I=2,NOP
      DO 2 J=I,NOP
      ARG=PHAZ*FLOAT(I-1)*FLOAT(J-1)
      S(I,J)=CMPLX(COS(ARG),SIN(ARG))
2     S(J,I)=S(I,J)
      GO TO 7
C
C     SET UP S MATRIX FOR PLANE SYMMETRY
C
3     KK=1
      S(1,1)=(1.,0.)
      IF ((NOP.EQ.2).OR.(NOP.EQ.4).OR.(NOP.EQ.8)) GO TO 4
      STOP
4     KA=NOP/2
      IF (NOP.EQ.8) KA=3
      DO 6 K=1,KA
      DO 5 I=1,KK
      DO 5 J=1,KK
      DETER=S(I,J)
      S(I,J+KK)=DETER
      S(I+KK,J+KK)=-DETER
5     S(I+KK,J)=DETER
6     KK=KK*2
C
C     COMBINE NP X NP SUBMATRICIES TO FORM MATRICIES OF SYMMETRY MODES.
C
7     IF (IC1.EQ.-1) GO TO 19
      I2=2*NPBLK*NROW
      ICOLS=NPBLK
      IF (ICASE.LT.3) ICOLS=NP
      DO 13 L=1,NBLOKS
      IF (ICASE.LT.3) GO TO 8
      CALL BLCKIN (11,1,I2,1,601)
      IF (L.EQ.NBLOKS) ICOLS=NLAST
8     CONTINUE
      DO 12 I=1,ICOLS
      DO 12 J=1,NP
      DO 9 K=1,NOP
      KA=J+(K-1)*NP
      D(K)=A(KA,I)
9     CONTINUE
      DETER=D(1)
      DO 10 KK=2,NOP
10    DETER=DETER+D(KK)
      A(J,I)=DETER
      DO 12 K=2,NOP
      KA=J+(K-1)*NP
      DETER=D(1)
      DO 11 KK=2,NOP
11    DETER=DETER+D(KK)*S(K,KK)
      A(KA,I)=DETER
12    CONTINUE
      IF (ICASE.LT.3) GO TO 13
      CALL BLCKOT (12,1,I2,1,603)
13    CONTINUE
14    IF (ICASE.GT.3) GO TO 17
C
C     FACTOR SUBMATRICIES, OR FACTOR COMPLETE MATRIX IF NO SYMMETRY
C     EXISTS.
C
      DO 16 KK=1,NOP
      KA=(KK-1)*NP+1
      IF (ICASE.EQ.3) GO TO 15
      CALL FACTR (NP,A(KA,1),IP(KA),NROW)
      GO TO 16
15    CALL FACIO (A,NROW,NCOL,IP)
```

```
      CALL LUNSCR (A,NROW,NCOL,IX,IP)
      IC3=-1
      CALL CHKPRT
16    CONTINUE
      RETURN
17    CONTINUE
C
C     REWRITE THE MATRICES BY COLUMNS ON TAPE 13
C
      DO 18 K=1,NOP
      REWIND 12
      ICOLS=NPBLK
      DO 18 L=1,NBLOKS
      CALL BLCKIN (12,1,12,1,602)
      IF (L.EQ.NBLOKS) ICOLS=NLAST
      DO 18 ICOLDX=1,ICOLS
      IR1=(K-1)*NP+1
      IR2=K*NP
      WRITE (13) (A(I,ICOLDX),I=IR1,IR2)
18    CONTINUE
19    CONTINUE
      IF (ICASE.EQ.5) GO TO 20
      IC1=-1
      CALL CHKPRT
20    REWIND 11
      REWIND 12
      REWIND 13
      REWIND 15
      IF (ICASE.EQ.5) GO TO 21
      CALL SUB1 (NP,A,NOP,IP,NROW)
      RETURN
21    CONTINUE
      CALL SUB2 (A,NP,NCOLS,NOP,IP,IX,NROW)
      RETURN
      END


      SUBROUTINE FBLOCK (NBLOKS,NPBLK,NLAST,IMAX,IROW,ICOL,INT)
C
C     FBLOCK DETERMINES BLOCK SIZE AND NUMBER OF BLOCKS WHEN OUT-OF-CORE
C     MATRIX STORAGE IS REQUIRED.  INT=0 RETURNED IF MATRIX FITS IN CORE
C
      IF (IROW*ICOL.LE.IMAX) GO TO 3
      INT=1
      NCOLMX=IMAX/IROW
      IF (NCOLMX.LE.1) STOP
      IF (NCOLMX.EQ.2) GO TO 1
      IF (NCOLMX/2.EQ.(NCOLMX-1)/2) NCOLMX=NCOLMX-1
1     NPBLK=NCOLMX/2
      NBLOKS=ICOL/NPBLK
      NLAST=ICOL-NBLOKS*NPBLK
      IF (NLAST.EQ.0) GO TO 2
      NBLOKS=NBLOKS+1
      GO TO 5
2     NLAST=NPBLK
      GO TO 5
3     NBLOKS=1
      NPBLK=0
      NLAST=0
      INT=0
4     RETURN
5     CONTINUE
      PRINT 6, NBLOKS,NPBLK,NLAST
      GO TO 4
C
6     FORMAT (1X,11H BLOCKING   ,4I5/)
      END


      SUBROUTINE FFLD (THET,PHI,ETH,EPH)
C
C     FFLD CALCULATES THE FAR ZONE RADIATED ELECTRIC FIELDS,
C     THE FACTOR EXP(J*K*R)/(R/LAMDA) NOT INCLUDED
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      COMMON /CRNT/ AIR(1000),AII(1000),BIR(1000),BII(1000),CIR(1000),CI
     1I(1000),CUR(2000)
      COMMON /GND/ ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,IPERF
      DIMENSION CAB(1), SAB(1)
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1))
      COMPLEX CIX,CIY,CIZ,EXA,ETH,EPH,CONST,CCX,CCY,CCZ,CDP,CUR
      COMPLEX ZRATI,ZRSIN,RRV,RRH,RRV1,RRH1,RRV2,RRH2,ZRATI2,TIX,TIY,TIZ
      COMPLEX STOR,T1,ZSCRN
      COMPLEX EX,EY,EZ,GX,GY,GZ
      DATA PI,TP,CONST/3.141592654,6.283185308,(0.,-29.979)/
      DATA STOR,ETA/(0.,2367.067),376.73/
      IF (IFAR.LT.4) GO TO 1
      T1=STOR/FLOAT(NRADL)
      T2=SCRWR*FLOAT(NRADL)
1     PHX=-SIN(PHI)
      PHY=COS(PHI)
      ROZ=COS(THET)
      ROZS=ROZ
      THX=ROZ*PHY
```

```
      THY=-ROZ*PHX
      THZ=-SIN(THET)
      ROX=-THZ*PHY
      ROY=THZ*PHX
      IF (N.EQ.0) GO TO 21
C
C     LOOP FOR STRUCTURE IMAGE IF ANY
C
      DO 20 K=1,KSYMP
C
C     CALCULATION OF REFLECTION COEFFECIENTS
C
      IF (K.EQ.1) GO TO 5
      IF (IPERF.NE.1) GO TO 2
C
C     FOR PERFECT GROUND
C
      RRV=(-1.,0.)
      RRH=(-1.,0.)
      GO TO 3
C
C     FOR INFINITE PLANAR GROUND
C
2     ZRSIN=CSQRT(1.-ZRATI*ZRATI*THZ*THZ)
      RRV=-(ROZ-ZRATI*ZRSIN)/(ROZ+ZRATI*ZRSIN)
      RRH=(ZRATI*ROZ-ZRSIN)/(ZRATI*ROZ+ZRSIN)
3     IF (IFAR.LE.1) GO TO 4
C
C     FOR THE CLIFF PROBLEM. TWO REFLCTION COEFFICIENTS CALCULATED
C
      RRV1=RRV
      RRH1=RRH
      TTHET=TAN(THET)
      IF (IFAR.EQ.4) GO TO 4
      ZRSIN=CSQRT(1.-ZRATI2*ZRATI2*THZ*THZ)
      RRV2=-(ROZ-ZRATI2*ZRSIN)/(ROZ+ZRATI2*ZRSIN)
      RRH2=(ZRATI2*ROZ-ZRSIN)/(ZRATI2*ROZ+ZRSIN)
      DARG=-TP*2.*CH*ROZ
4     ROZ=-ROZ
      CCX=CIX
      CCY=CIY
      CCZ=CIZ
5     CIX=(0.,0.)
      CIY=(0.,0.)
      CIZ=(0.,0.)
C
C     LOOP OVER STRUCTURE SEGMENTS
C
      DO 18 I=1,N
      OMEGA=-(ROX*CAB(I)+ROY*SAB(I)+ROZ*SALP(I))
      EL=PI*SI(I)
      SILL=OMEGA*EL
      TOP=EL+SILL
      BOT=EL-SILL
      IF (ABS(OMEGA).LT.1.E-7) GO TO 6
      A=2.*SIN(SILL)/OMEGA
      GO TO 7
6     A=(2.-OMEGA*OMEGA*EL*EL/3.)*EL
7     IF (ABS(TOP).LT.1.E-7) GO TO 8
      TOO=SIN(TOP)/TOP
      GO TO 9
8     TOO=1.-TOP*TOP/6.
9     IF (ABS(BOT).LT.1.E-7) GO TO 10
      BOO=SIN(BOT)/BOT
      GO TO 11
10    BOO=1.-BOT*BOT/6.
11    B=EL*(BOO-TOO)
      C=EL*(BOO+TOO)
      RR=A*AIR(I)+B*BII(I)+C*CIR(I)
      RI=A*AII(I)-B*BIR(I)+C*CII(I)
      ARG=TP*(X(I)*ROX+Y(I)*ROY+Z(I)*ROZ)
      IF (K.EQ.2.AND.IFAR.GE.2) GO TO 12
      EXA=CMPLX(COS(ARG),SIN(ARG))*CMPLX(RR,RI)
C
C     SUMMATION FOR FAR FIELD INTEGRAL
C
      CIX=CIX+EXA*CAB(I)
      CIY=CIY+EXA*SAB(I)
      CIZ=CIZ+EXA*SALP(I)
      GO TO 18
C
C     CALCULATION OF IMAGE CONTRIBUTION IN CLIFF AND GROUND SCREEN
C     PROBLEMS.
C
12    DR=Z(I)*TTHET
C
C     SPECULAR POINT DISTANCE
C
      D=DR*PHY+X(I)
      IF (IFAR.EQ.2) GO TO 14
      D=SQRT(D*D+(Y(I)-DR*PHX)**2)
      IF (IFAR.EQ.3) GO TO 14
      IF ((SCRWL-D).LT.0.) GO TO 13
C
C     RADIAL WIRE GROUND SCREEN REFLECTION COEFFICIENT
C
      D=D+T2
      ZSCRN=T1*D*ALOG(D/T2)
      ZSCRN=(ZSCRN*ZRATI)/(ETA*ZRATI+ZSCRN)
```

```
      ZRSIN=CSQRT(1.-ZSCRN*ZSCRN*THZ*THZ)
      RRV=(ROZ+ZSCRN*ZRSIN)/(-ROZ+ZSCRN*ZRSIN)
      RRH=(ZSCRN*ROZ+ZRSIN)/(ZSCRN*ROZ-ZRSIN)
      GO TO 17
13    IF (IFAR.EQ.4) GO TO 15
      IF (IFAR.EQ.5) D=DR*PHY+X(I)
14    IF ((CL-D).LE.0.) GO TO 16
15    RRV=RRV1
      RRH=RRH1
      GO TO 17
16    RRV=RRV2
      RRH=RRH2
      ARG=ARG+DARG
17    EXA=CMPLX(COS(ARG),SIN(ARG))*CMPLX(RR,RI)
C
C     CONTRIBUTION OF EACH IMAGE SEGMENT MODIFIED BY REFLECTION COEF. ,
C     FOR CLIFF AND GROUND SCREEN PROBLEMS
C
      TIX=EXA*CAB(I)
      TIY=EXA*SAB(I)
      TIZ=EXA*SALP(I)
      CDP=(TIX*PHX+TIY*PHY)*(RRH-RRV)
      CIX=CIX+TIX*RRV+CDP*PHX
      CIY=CIY+TIY*RRV+CDP*PHY
      CIZ=CIZ-TIZ*RRV
18    CONTINUE
      IF (K.EQ.1) GO TO 20
      IF (IFAR.GE.2) GO TO 19
C
C     CALCULATION OF CONTRIBUTION OF STRUCTURE IMAGE FOR INFINITE GROUND
C
      CDP=(CIX*PHX+CIY*PHY)*(RRH-RRV)
      CIX=CCX+CIX*RRV+CDP*PHX
      CIY=CCY+CIY*RRV+CDP*PHY
      CIZ=CCZ-CIZ*RRV
      GO TO 20
19    CIX=CIX+CCX
      CIY=CIY+CCY
      CIZ=CIZ+CCZ
20    CONTINUE
      IF (M.GT.0) GO TO 22
      ETH=(CIX*THX+CIY*THY+CIZ*THZ)*CONST
      EPH=(CIX*PHX+CIY*PHY)*CONST
      RETURN
21    CIX=(0.,0.)
      CIY=(0.,0.)
      CIZ=(0.,0.)
22    ROZ=ROZS
C
C     ELECTRIC FIELD COMPONENTS DUE TO SURFACE CURRENTS
C
      EX=(0.,0.)
      EY=(0.,0.)
      EZ=(0.,0.)
      RFL=-1.
      DO 23 IP=1,KSYMP
      RFL=-RFL
      RRZ=ROZ*RFL
      CALL FFLDS (ROX,ROY,RRZ,CUR(N+1),GX,GY,GZ)
      EX=EX+GX*RFL
      EY=EY+GY*RFL
23    EZ=EZ+GZ
      EX=EX+CIX*CONST
      EY=EY+CIY*CONST
      EZ=EZ+CIZ*CONST
      ETH=EX*THX+EY*THY+EZ*THZ
      EPH=EX*PHX+EY*PHY
      RETURN
      END


      SUBROUTINE FFLDS (ROX,ROY,ROZ,SCUR,EX,EY,EZ)
C
C     CALCULATES THE XYZ COMPONENTS OF THE ELECTRIC FIELD DUE TO
C     SURFACE CURRENTS IN ARRAY SCUR
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION SCUR(1)
      DIMENSION XS(1), YS(1), ZS(1), S(1)
      COMPLEX CT,CONS,SCUR,EX,EY,EZ
      EQUIVALENCE (XS,X), (YS,Y), (ZS,Z), (S,BI)
      DATA CONS/(0.,188.365)/
      DATA TPI/6.283185308/
      EX=(0.,0.)
      EY=(0.,0.)
      EZ=(0.,0.)
      I=LD+1
      DO 1 J=1,M
      I=I-1
      ARG=TPI*(ROX*XS(I)+ROY*YS(I)+ROZ*ZS(I))
      CT=CMPLX(COS(ARG)*S(I),SIN(ARG)*S(I))
      K=3*J
      EX=EX+SCUR(K-2)*CT
      EY=EY+SCUR(K-1)*CT
      EZ=EZ+SCUR(K)*CT
1     CONTINUE
```

```
      CT=ROX*EX+ROY*EY+ROZ*EZ
      EX=CONS*(CT*ROX-EX)
      EY=CONS*(CT*ROY-EY)
      EZ=CONS*(CT*ROZ-EZ)
      RETURN
      END
```

```
      SUBROUTINE GF (ZK,CO,SI)
C
C     GF COMPUTES THE INTEGRAND EXP(JKR)/(KR) FOR NUMERICAL INTEGRATION.
C
      COMMON /TMI/ ZPK,RKB2,IJ
      ZDK=ZK-ZPK
      RK=SQRT(RKB2+ZDK*ZDK)
      SI=SIN(RK)/RK
      IF (IJ) 1,2,1
1     CO=COS(RK)/RK
      RETURN
2     IF (RK.LT..2) GO TO 3
      CO=(COS(RK)-1.)/RK
      RETURN
3     RKS=RK*RK
      CO=((-1.38888889E-3*RKS+4.16666667E-2)*RKS-.5)*RK
      RETURN
      END
```

```
      SUBROUTINE GFLD (RHO,PHI,RZX,ETH,EPI,ERD,UX,KSYMP)
C
C     GFLD COMPUTES THE RADIATED FIELD INCLUDING GROUND WAVE.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      COMMON /CRNT/ AIR(1000),AII(1000),BIR(1000),BII(1000),CIR(1000),CI
     1I(1000),CUR(2000)
      COMMON /GWAV/ RHS,CPH,SPH,RZ,H,U,U2,XX1,XX2,ERV,EZV,ERH,EPH,EZH
      DIMENSION CAB(1), SAB(1)
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1))
      COMPLEX CUR,EPI,CIX,CIY,CIZ,EXA,XX1,XX2,U,U2,ERV,EZV,ERH,EPH
      COMPLEX EZH,EX,EY,ETH,UX,ERD
      DATA PI,TP/3.1415926,6.2831853/
      R=SQRT(RHO*RHO+RZX*RZX)
      IF (KSYMP.EQ.1) GO TO 1
      IF (CABS(UX).GT..5) GO TO 1
      IF (R.GT.1.E5) GO TO 1
      GO TO 4
C
C     COMPUTATION OF SPACE WAVE ONLY
C
1     IF (RZX.LT.1.E-20) GO TO 2
      THET=ATAN(RHO/RZX)
      GO TO 3
2     THET=PI*.5
3     CALL FFLD (THET,PHI,ETH,EPI)
      ARG=-TP*R
      EXA=CMPLX(COS(ARG),SIN(ARG))/R
      ETH=ETH*EXA
      EPI=EPI*EXA
      ERD=(0.,0.)
      RETURN
C
C     COMPUTATION OF SPACE AND GROUND WAVES.
C
4     RZ=RZX
      U=UX
      U2=U*U
      PHX=-SIN(PHI)
      PHY=COS(PHI)
      RX=RHO*PHY
      RY=-RHO*PHX
      CIX=(0.,0.)
      CIY=(0.,0.)
      CIZ=(0.,0.)
C
C     SUMMATION OF FIELD FROM INDIVIDUAL SEGMENTS
C
      DO 17 I=1,N
      DX=CAB(I)
      DY=SAB(I)
      DZ=SALP(I)
      RIX=RX-X(I)
      RIY=RY-Y(I)
      RHS=RIX*RIX+RIY*RIY
      RHP=SQRT(RHS)
      IF (RHP.LT.1.E-6) GO TO 5
      RHX=RIX/RHP
      RHY=RIY/RHP
      GO TO 6
5     RHX=1.
      RHY=0.
6     CALP=1.-DZ*DZ
      IF (CALP.LT.1.E-6) GO TO 7
      CALP=SQRT(CALP)
      CBET=DX/CALP
```

```
      SBET=DY/CALP
      CPH=RHX*CBET+RHY*SBET
      SPH=RHY*CBET-RHX*SBET
      GO TO 8
7     CPH=RHX
      SPH=RHY
8     EL=PI*SI(I)
      RFL=-1.
C
C     INTEGRATION OF (CURRENT)*(PHASE FACTOR) OVER SEGMENT AND IMAGE FOR
C     CONSTANT, SINE, AND COSINE CURRENT DISTRIBUTIONS
C
      DO 16 K=1,2
      RFL=-RFL
      RIZ=RZ-Z(I)*RFL
      RXYZ=SQRT(RIX*RIX+RIY*RIY+RIZ*RIZ)
      RNX=RIX/RXYZ
      RNY=RIY/RXYZ
      RNZ=RIZ/RXYZ
      OMEGA=-(RNX*DX+RNY*DY+RNZ*DZ*RFL)
      SILL=OMEGA*EL
      TOP=EL+SILL
      BOT=EL-SILL
      IF (ABS(OMEGA).LT.1.E-7) GO TO 9
      A=2.*SIN(SILL)/OMEGA
      GO TO 10
9     A=(2.-OMEGA*OMEGA*EL*EL/3.)*EL
10    IF (ABS(TOP).LT.1.E-7) GO TO 11
      TOO=SIN(TOP)/TOP
      GO TO 12
11    TOO=1.-TOP*TOP/6.
12    IF (ABS(BOT).LT.1.E-7) GO TO 13
      BOO=SIN(BOT)/BOT
      GO TO 14
13    BOO=1.-BOT*BOT/6.
14    B=EL*(BOO-TOO)
      C=EL*(BOO+TOO)
      RR=A*AIR(I)+B*BII(I)+C*CIR(I)
      RI=A*AII(I)-B*BIR(I)+C*CII(I)
      ARG=TP*(X(I)*RNX+Y(I)*RNY+Z(I)*RNZ*RFL)
      EXA=CMPLX(COS(ARG),SIN(ARG))*CMPLX(RR,RI)/TP
      IF (K.EQ.2) GO TO 15
      XX1=EXA
      GO TO 16
15    XX2=EXA
16    CONTINUE
      H=Z(I)
C
C     CALL SUBROUTINE TO COMPUTE THE FIELD OF SEGMENT INCLUDING GROUND
C     WAVE.
C
      CALL GWAVE
      ERH=ERH*CALP+ERV*DZ
      EPH=EPH*CALP
      EZH=EZH*CALP+EZV*DZ
      EX=ERH*RHX-EPH*RHY
      EY=ERH*RHY+EPH*RHX
      CIX=CIX+EX
      CIY=CIY+EY
17    CIZ=CIZ+EZH
      ARG=-TP*R
      EXA=CMPLX(COS(ARG),SIN(ARG))
      CIX=CIX*EXA
      CIY=CIY*EXA
      CIZ=CIZ*EXA
      RNX=RX/R
      RNY=RY/R
      RNZ=RZ/R
      THX=RNZ*PHY
      THY=-RNZ*PHX
      THZ=-RHO/R
      ETH=CIX*THX+CIY*THY+CIZ*THZ
      EPI=CIX*PHX+CIY*PHY
      ERD=CIX*RNX+CIY*RNY+CIZ*RNZ
      RETURN
      END


      SUBROUTINE GH (ZK,HR,HI)
C
C     COMPUTES INTEGRAND FOR NUMERICAL INTEGRATION OF MAGNETIC FIELD DUE
C     TO A CONSTANT CURRENT ON A WIRE SEGMENT
C
      COMMON /TMH/ ZPK,RHKS
      RS=ZK-ZPK
      RS=RHKS+RS*RS
      R=SQRT(RS)
      CKR=COS(R)
      SKR=SIN(R)
      RR2=1./RS
      RR3=RR2/R
      HR=SKR*RR2+CKR*RR3
      HI=CKR*RR2-SKR*RR3
      RETURN
      END
```

```
      SUBROUTINE GN (EZR,EZI,ERR,ERI)
C
C     GN MULTIPLIES THE FIELD COMPONENTS IN AND NORMAL TO THE PLANE OF
C     INCIDENCE BY THEIR GROUND PLANE REFLECTION COEFFICIENTS TO RETURN
C     TOTAL FIELD AFTER REFLECTION.
C
      COMMON /REFL/ RHOX,RHOY,RHOZ,CABJ,SABJ,SALPR,PX,PY,REFS,REFPS
      COMMON /GND/ ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,IPERF
      COMPLEX EZ,ER,ERX,ERY,ERZ,EPX,EPY,REFS,REFPS,ZRATI,ZRATI2
      IF (IPERF.NE.1) GO TO 1
      EZR=-EZR
      EZI=-EZI
      ERR=-ERR
      ERI=-ERI
      RETURN
1     EZ=CMPLX(EZR,EZI)
      ER=CMPLX(ERR,ERI)
      ERX=RHOX*ER+CABJ*EZ
      ERY=RHOY*ER+SABJ*EZ
      ERZ=RHOZ*ER+SALPR*EZ
      EPY=PX*ERX+PY*ERY
      EPX=PX*EPY
      EPY=PY*EPY
      ERX=REFS*ERX+REFPS*EPX
      ERY=REFS*ERY+REFPS*EPY
      ERZ=REFS*ERZ
      EZ=ERX*CABJ+ERY*SABJ+ERZ*SALPR
      ER=ERX*RHOX+ERY*RHOY+ERZ*RHOZ
      EZR=REAL(EZ)
      EZI=AIMAG(EZ)
      ERR=REAL(ER)
      ERI=AIMAG(ER)
      RETURN
      END
```

```
      SUBROUTINE GWAVE
C
C     GWAVE COMPUTES THE ELECTRIC FIELD, INCLUDING GROUND WAVE, OF A
C     CURRENT ELEMENT OVER A GROUND PLANE USING FORMULAS OF K.A. NORTON
C     (PROC. IRE, SEPT., 1937, PP.1203-1236.)
C
      COMMON /GWAV/ RHS,CPH,SPH,RZ,H,U,U2,XX1,XX2,ERV,EZV,ERH,EPH,EZH
      COMPLEX FJ,TPJ,U2,U,RK1,RK2,T1,T2,T3,T4,P1,RV,OMR,W,F,Q1,RH,V,G,XR
     11,XR2,X1,X2,X3,X4,X5,X6,X7,EZV,ERV,EZH,ERH,EPH,XX1,XX2,ECON,EFUN
      DATA FJ/(0.,1.)/,TPJ/(0.,6.283185 3)/
      DATA PI/3.1415926/,ECON/(0.,-188.363)/
      ZD=RZ-H
      R1=SQRT(RHS+ZD*ZD)
      SPPP=ZD/R1
      SPPP2=SPPP*SPPP
      CPPP2=1.-SPPP2
      CPPP=SQRT(CPPP2)
      ZD=RZ+H
      R2=SQRT(RHS+ZD*ZD)
      SPP=ZD/R2
      SPP2=SPP*SPP
      CPP2=1.-SPP2
      CPP=SQRT(CPP2)
      RK1=-TPJ*R1
      RK2=-TPJ*R2
      T1=1.-U2*CPP2
      T2=CSQRT(T1)
      T3=(1.-1./RK1)/RK1
      T4=(1.-1./RK2)/RK2
      P1=RK2*U2*T1*.5
      RV=(SPP-U*T2)/(SPP+U*T2)
      OMR=1.-RV
      W=1./OMR
      W=(4.,0.)*P1*W*W
      F=FJ*CSQRT(W)
      F=1.-EFUN(F)
      Q1=RK2*T1/(2.*U2)
      RH=(T2-U*SPP)/(T2+U*SPP)
      V=1./(1.+RH)
      V=(4.,0.)*Q1*V*V
      G=FJ*CSQRT(V)
      G=1.-EFUN(G)
      XR1=XX1/R1
      XR2=XX2/R2
      X1=CPPP2*XR1
      X2=RV*CPP2*XR2
      X3=OMR*(1.-U2+U2*U2*CPP2)*F*XR2
      X4=U*T2*SPP*2.*XR2/RK2
      X5=XR1*T3*(1.-3.*SPPP2)
      X6=XR2*T4*(1.-3.*SPP2)
      EZV=(X1+X2+X3-X4-X5-X6)*ECON
      X1=SPPP*CPPP*XR1
      X2=RV*SPP*CPP*XR2
      X3=CPP*OMR*U*T2*F*XR2*(1.-.5*(U2*T1-SPP2+1./RK2))
      X4=SPP*CPP*OMR*XR2/RK2
      X5=3.*SPPP*CPPP*T3*XR1
      X6=CPP*U*T2*OMR*XR2/RK2*.5
      X7=3.*SPP*CPP*T4*XR2
      ERV=-(X1+X2-X3+X4-X5+X6-X7)*ECON
      EZH=-CPH*(X1-X2+X3-X4-X5-X6+X7)*ECON
      X1=SPPP2*XR1
      X2=RV*SPP2*XR2
```

```
      X3=(1.+RH)*G*XR2
      X4=CPP2*U2*OMR*F*XR2
      X5=T3*(1.-3.*CPPP2)*XR1
      X6=T4*(1.-3.*CPP2)*(1.-U2*(1.+RV)-U2*OMR*F)*XR2
      X7=U2*CPP2*OMR*(1.-1./RK2)*(F*(U2*T1-SPP2-1./RK2)+1./RK2)*XR2
      ERH=CPH*(X1-X2+X3-X4-X5+X6+X7)*ECON
      X1=XR1
      X2=RH*XR2
      X3=(RH+1.)*G*XR2
      X4=T3*XR1
      X5=T4*(1.-U2*(1.+RV)-U2*OMR*F)*XR2
      X6=.5*U2*OMR*(F*(U2*T1-SPP2-1./RK2)+1./RK2)*XR2/RK2
      EPH=-SPH*(X1-X2+X3-X4+X5+X6)*ECON
      RETURN
      END


      SUBROUTINE HFK (EL1),EL2,RHK,ZPKX,SGR,SGI)
C
C     HFK PERFORMS NUMERICAL INTEGRATION FOR THE MAGNETIC FIELD OF A
C     CONSTANT CURRENT ON A WIRE SEGMENT BY THE METHOD OF VARIABLE
C     INTERVAL WIDTH ROMBERG INTEGRATION.
C
      COMMON /TMH/ ZPK,RHKS
      DATA NX,NM,NTS,RX/1,65536,4,1.E-4/
      ZPK=ZPKX
      RHKS=RHK*RHK
      Z=EL1
      ZE=EL2
      S=ZE-Z
      EP=S/(10.*NM)
      ZEND=ZE-EP
      SGR=0.0
      SGI=0.0
      NS=NX
      NT=0
      CALL GH (Z,G1R,G1I)
1     DZ=S/NS
      DZOT=DZ*0.5
      ZP=Z+DZ
      IF (ZP-ZE) 3,3,2
2     DZ=ZE-Z
      IF (ABS(DZ)-EP) 17,17,3
3     ZP=Z+DZOT
      CALL GH (ZP,G3R,G3I)
      ZP=Z+DZ
      CALL GH (ZP,G5R,G5I)
4     T00R=(G1R+G5R)*DZOT
      T00I=(G1I+G5I)*DZOT
      T01R=(T00R+DZ*G3R)*0.5
      T01I=(T00I+DZ*G3I)*0.5
      T10R=(4.0*T01R-T00R)/3.0
      T10I=(4.0*T01I-T00I)/3.0
      CALL TEST (T01R,T10R,TE1R,T01I,T10I,TE1I)
      IF (TE1I-RX) 5,5,6
5     IF (TE1R-RX) 8,8,6
6     ZP=Z+DZ*0.25
      CALL GH (ZP,G2R,G2I)
      ZP=Z+DZ*0.75
      CALL GH (ZP,G4R,G4I)
      T02R=(T01R+DZOT*(G2R+G4R))*0.5
      T02I=(T01I+DZOT*(G2I+G4I))*0.5
      T11R=(4.0*T02R-T01R)/3.0
      T11I=(4.0*T02I-T01I)/3.0
      T20R=(16.0*T11R-T10R)/15.0
      T20I=(16.0*T11I-T10I)/15.0
      CALL TEST (T11R,T20R,TE2R,T11I,T20I,TE2I)
      IF (TE2I-RX) 7,7,14
7     IF (TE2R-RX) 9,9,14
8     SGR=SGR+T10R
      SGI=SGI+T10I
      NT=NT+2
      GO TO 10
9     SGR=SGR+T20R
      SGI=SGI+T20I
      NT=NT+1
10    Z=Z+DZ
      IF (Z-ZEND) 11,17,17
11    G1R=G5R
      G1I=G5I
      IF (NT-NTS) 1,12,12
12    IF (NS-NX) 1,1,13
13    NS=NS/2
      NT=1
      GO TO 1
14    NT=0
      IF (NS-NM) 16,15,15
15    PRINT 18, Z
      GO TO 9
16    NS=NS*2
      DZ=S/NS
      DZOT=DZ*0.5
      G5R=G3R
      G5I=G3I
      G3R=G2R
      G3I=G2I
      GO TO 4
17    CONTINUE
```

```
      SGR=SGR*RHK*.5
      SGI=SGI*RHK*.5
      RETURN
C
18    FORMAT (24H STEP SIZE LIMITED AT Z=,F10.5)
      END


      SUBROUTINE HFLD (S,RH,ZP,HPS,HPC,HPK)
C
C     CALCULATES H FIELD OF SINE COSINE, AND CONSTANT CURRENT OF SEGMENT
C
      COMPLEX FJ,FJK,EKR1,EKR2,T1,T2,CONS,HPS,HPC,HPK
      DATA TP,FJ,FJK/6.28318530B,(0.,1.),(0.,-6.28318530B)/
      IF (RH.LT.1.E-20) GO TO 1
      RH2=RH*RH
      DH=S*.5
      DK=TP*DH
      CDK=COS(DK)
      SDK=SIN(DK)
      Z1=ZP+DH
      Z2=ZP-DH
      R1=SQRT(RH2+Z1*Z1)
      R2=SQRT(RH2+Z2*Z2)
      EKR1=CEXP(FJK*R1)
      EKR2=CEXP(FJK*R2)
      T1=Z1*EKR1/R1
      T2=Z2*EKR2/R2
      HPS=CDK*(EKR2-EKR1)-FJ*SDK*(T2+T1)
      HPC=-SDK*(EKR2+EKR1)-FJ*CDK*(T2-T1)
      CALL HFK (-DK,DK,RH*TP,ZP*TP,HKR,HKI)
      CONS=-FJ/(2.*TP*RH)
      HPS=CONS*HPS
      HPC=CONS*HPC
      HPK=CMPLX(HKR,HKI)
      RETURN
1     HPS=(0.,0.)
      HPC=(0.,0.)
      HPK=(0.,0.)
      RETURN
      END


      SUBROUTINE HINTG (II,JJ,G11,G12,G21,G22,IP)
C
C     COMPUTE H FIELD DUE TO A SINGLE PATCH
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      COMMON /DATI/ T1XI,T1YI,T1ZI,T2XI,T2YI,T2ZI,XI,YI,ZI
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), S(1)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG), (S,BI)
      COMPLEX GAM,GX,GY,GZ,F1X,F1Y,F1Z,F2X,F2Y,F2Z,G11,G12,G21,G22
      DATA FPI/12.566370616/,TPI/6.283185308/
      IF (II.EQ.JJ.AND.IP.EQ.1) RETURN
      RFL=FLOAT(3-2*IP)
      J=LD+1-JJ
      RX=XI-X(J)
      RY=YI-Y(J)
      RZ=ZI-Z(J)*RFL
      RSQ=RX*RX+RY*RY+RZ*RZ
      R=SQRT(RSQ)
      RK=-TPI*R
      CR=COS(RK)
      SR=SIN(RK)
      GAM=(CMPLX(CR,SR)+RK*CMPLX(SR,-CR))/(FPI*RSQ*R)*S(J)
      GX=GAM*RX
      GY=GAM*RY
      GZ=GAM*RZ
      T1QX=T1X(J)
      T1QY=T1Y(J)
      T1QZ=T1Z(J)*RFL
      T2QX=T2X(J)
      T2QY=T2Y(J)
      T2QZ=T2Z(J)*RFL
      F1X=GY*T1QZ-GZ*T1QY
      F1Y=GZ*T1QX-GX*T1QZ
      F1Z=GX*T1QY-GY*T1QX
      F2X=GY*T2QZ-GZ*T2QY
      F2Y=GZ*T2QX-GX*T2QZ
      F2Z=GX*T2QY-GY*T2QX
      G11=(T2XI*F1X+T2YI*F1Y+T2ZI*F1Z)*RFL+G11
      G12=(T2XI*F2X+T2YI*F2Y+T2ZI*F2Z)*RFL+G12
      G21=(T1XI*F1X+T1YI*F1Y+T1ZI*F1Z)*RFL+G21
      G22=(T1XI*F2X+T1YI*F2Y+T1ZI*F2Z)*RFL+G22
      RETURN
      END


      SUBROUTINE HMAT (IM1,IM2,CM,NROW,NCOL)
C
C     HMAT FILLS THE MATRIX ELEMENTS REPRESENTING THE H FIELD ON PATCHES
C     DUE TO SURFACE CURRENTS ON OTHER PATCHES
```

```
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      COMMON /DATI/ T1XI,T1YI,T1ZI,T2XI,T2YI,T2ZI,XI,YI,ZI
      COMMON /GND/ ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,IPERF
      DIMENSION CM(NROW,NCOL)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), XS(1), Y
     1S(1), ZS(1)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG), (XS,X), (YS,Y), (ZS,Z)
      COMPLEX G11,G12,G21,G22,CM
      COMPLEX ZRATI,ZRATI2
      ISTART=IM1
      IF (IM1.LE.NP) ISTART=NP+1
      IPST=(ISTART-NP+1)/2
      IPEND=(IM2-NP+1)/2
      ISELST=(ISTART-NP)/IPST
      ISELEN=(IM2-NP)/IPEND
      K=ISTART-IM1-1
      IF (ISELST.EQ.2) K=K-1
      NOP=M/MP
      IL=LD-IPST+2
C     OBSERVATION LOOP
      DO 7 IP=IPST,IPEND
      K=K+2
      I1=K
      I2=K+1
      IL=IL-1
      T1XI=T1X(IL)*SALP(IL)
      T1YI=T1Y(IL)*SALP(IL)
      T1ZI=T1Z(IL)*SALP(IL)
      T2XI=T2X(IL)*SALP(IL)
      T2YI=T2Y(IL)*SALP(IL)
      T2ZI=T2Z(IL)*SALP(IL)
      XI=XS(IL)
      YI=YS(IL)
      ZI=ZS(IL)
C     SOURCE LOOP
      J=0
      DO 4 L=1,NOP
      J2=L*(NP+2*MP)-2*MP
      DO 4 LL=1,MP
      J2=J2+2
      J1=J2-1
      J=J+1
      G11=0.
      G12=0.
      G21=0.
      G22=0.
      DO 1 IX=1,KSYMP
1     CALL HINTG (IP,J,G11,G12,G21,G22,IX)
      IF (IP.NE.IPST.OR.ISELST.EQ.1) GO TO 2
      CM(J1,I2)=G21
      CM(J2,I2)=G22
      GO TO 4
2     IF (IP.NE.IPEND.OR.ISELEN.EQ.2) GO TO 3
      CM(J1,I1)=G11
      CM(J2,I1)=G12
      GO TO 4
3     CM(J1,I1)=G11
      CM(J2,I1)=G12
      CM(J1,I2)=G21
      CM(J2,I2)=G22
4     CONTINUE
      J2=((IP-1)/MP+1)*NP+2*IP
      J1=J2-1
      IF (IP.NE.IPST.OR.ISELST.EQ.1) GO TO 5
      CM(J2,I2)=.5+CM(J2,I2)
      GO TO 7
5     IF (IP.NE.IPEND.OR.ISELEN.EQ.2) GO TO 6
      CM(J1,I1)=-.5+CM(J1,I1)
      GO TO 7
6     CONTINUE
      CM(J1,I1)=-.5+CM(J1,I1)
      CM(J2,I2)=0.5+CM(J2,I2)
7     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HWMAT (XJ,YJ,ZT,SJ,CABJ,SABJ,SALPT,DIL,DIK,IPATCH,TWHR,
     1TWHI,IP,RKH,IFLG)
C
C     HWMAT COMPUTES THE MATRIX ELEMENTS REPRESENTING THE H FIELD AT A
C     PATCH DUE TO THE CURRENT ON A WIRE SEGMENT.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      DIMENSION TWHR(3,2), TWHI(3,2)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), XS(1), Y
     1S(1), ZS(1)
      COMPLEX HPS,HPC,HPK,HSDT,HCDT,HKDT
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG), (XS,X), (YS,Y), (ZS,Z)
```

```
      DATA TP/6.283185308/
      RFL=FLOAT(3-2*IP)
      ZJ=ZT*RFL
      SALPJ=SALPT*RFL
      ILC=LD+1-IPATCH
      XD=XS(ILC)-XJ
      YD=YS(ILC)-YJ
      ZD=ZS(ILC)-ZJ
      ZP=XD*CABJ+YD*SABJ+ZD*SALPJ
      RHX=XD-CABJ*ZP
      RHY=YD-SABJ*ZP
      RHZ=ZD-SALPJ*ZP
      RH=SQRT(RHX*RHX+RHY*RHY+RHZ*RHZ)
      IF (RH.LT.1.E-20) GO TO 3
      RHX=RHX/RH
      RHY=RHY/RH
      RHZ=RHZ/RH
      PX=SABJ*RHZ-SALPJ*RHY
      PY=SALPJ*RHX-CABJ*RHZ
      PZ=CABJ*RHY-SABJ*RHX
      R2=XD*XD+YD*YD+ZD*ZD
      R=SQRT(R2)
      IF (R.GT.RKH) GO TO 2
      IFLG=1
      CALL HFLD (SJ,RH,ZP,HPS,HPC,HPK)
      HPS=HPS*RFL
      HPC=HPC*RFL
      HPK=HPK*RFL
      CL=TP*DIL
      CK=TP*DIK
      SINL=SIN(CL)
      COSL=COS(CL)
      SINK=SIN(CK)
      COSK=COS(CK)
      SILK=SIN(CL+CK)
      CONS=SINL+SINK-SILK
      DO 1 K=1,2
      IF (K.EQ.1) PDT=PX*T2X(ILC)+PY*T2Y(ILC)+PZ*T2Z(ILC)
      IF (K.EQ.2) PDT=PX*T1X(ILC)+PY*T1Y(ILC)+PZ*T1Z(ILC)
      PDT=PDT*SALP(ILC)
      HSDT=-HPS*PDT
      HCDT=-HPC*PDT
      HKDT=-HPK*PDT
      TWHR(1,K)=(SINK*REAL(HKDT)+(COSK-1.)*REAL(HSDT)-SINK*REAL(HCDT))/C
     1ONS+TWHR(1,K)
      TWHI(1,K)=(SINK*AIMAG(HKDT)+(COSK-1.)*AIMAG(HSDT)-SINK*AIMAG(HCDT)
     1)/CONS+TWHI(1,K)
      TWHR(2,K)=(-SILK*REAL(HKDT)+(COSL-COSK)*REAL(HSDT)+(SINL+SINK)*REA
     1L(HCDT))/CONS+TWHR(2,K)
      TWHI(2,K)=(-SILK*AIMAG(HKDT)+(COSL-COSK)*AIMAG(HSDT)+(SINL+SINK)*A
     1IMAG(HCDT))/CONS+TWHI(2,K)
      TWHR(3,K)=(SINL*REAL(HKDT)+(1.-COSL)*REAL(HSDT)-SINL*REAL(HCDT))/C
     1ONS+TWHR(3,K)
      TWHI(3,K)=(SINL*AIMAG(HKDT)+(1.-COSL)*AIMAG(HSDT)-SINL*AIMAG(HCDT)
     1)/CONS+TWHI(3,K)
1     CONTINUE
      RETURN
2     CALL APRXH (R,R2,SJ,TP,RH/R,TWHR,TWHI,ILC,PX,PY,PZ,RFL)
      RETURN
3     IF (ZP.LT.1.E-20) GO TO 4
      RETURN
4     PRINT 5
      STOP
C
5     FORMAT (46H H FIELD AT SOURCE POINT IS UNDEFINED IN HWMAT)
      END


      SUBROUTINE INTG (B,S,RH,ZP,DIJ,DIR,ETR,ETI,DIL,DIK,IJ,IP)
C
C     INTG COMPUTES THREE COMPLEX FIELD COMPONENTS THAT MULTIPLY THE
C     THREE SEGMENT CURRENTS USED IN INTERPOLATING OVER A SEGMENT.
C     THESE COMPONENTS, ETR AND ETI, GO INTO THE INTERACTION MATRIX.
C
      DIMENSION ETR(3), ETI(3)
      DATA TP/6.283185308/
C
C     COMPUTE TANGENTIAL FIELD ON OBSERVATION SEGMENT DUE TO SINE,
C     COSINE, AND CONSTANT CURRENTS ON SOURCE SEGMENT.
C
      CALL EFLD (B,S,RH,ZP,IJ,EZRS,EZIS,ERRS,ERIS,EZRC,EZIC,ERRC,ERIC,EZ
     1RK,EZIK,ERRK,ERIK)
      IF (IP.NE.2) GO TO 1
      CALL GN (EZRS,EZIS,ERRS,ERIS)
      CALL GN (EZRC,EZIC,ERRC,ERIC)
      CALL GN (EZRK,EZIK,ERRK,ERIK)
1     ETRS=EZRS*DIJ+ERRS*DIR
      ETIS=EZIS*DIJ+ERIS*DIR
      ETRC=EZRC*DIJ+ERRC*DIR
      ETIC=EZIC*DIJ+ERIC*DIR
      ETRK=EZRK*DIJ+ERRK*DIR
      ETIK=EZIK*DIJ+ERIK*DIR
C
C     COMPUTE INTERPOLATION COEFFICIENTS AND FORM THE COEFFICIENTS OF
C     THE THREE SEGMENT CURRENTS USED IN CURRENT INTERPOLATION.
C
      CL=TP*DIL
      CK=TP*DIK
```

```
      SINL=SIN(CL)
      COSL=COS(CL)
      SINK=SIN(CK)
      COSK=COS(CK)
      SILK=SIN(CL+CK)
      CONS=SINL+SINK-SILK
      ETR(1)=(SINK*ETRK+(COSK-1.)*ETRS-SINK*ETRC)/CONS+ETR(1)
      ETI(1)=(SINK*ETIK+(COSK-1.)*ETIS-SINK*ETIC)/CONS+ETI(1)
      ETR(2)=(-SILK*ETRK+(COSL-COSK)*ETRS+(SINL+SINK)*ETRC)/CONS+ETR(2)
      ETI(2)=(-SILK*ETIK+(COSL-COSK)*ETIS+(SINL+SINK)*ETIC)/CONS+ETI(2)
      ETR(3)=(SINL*ETRK+(1.-COSL)*ETRS-SINL*ETRC)/CONS+ETR(3)
      ETI(3)=(SINL*ETIK+(1.-COSL)*ETIS-SINL*ETIC)/CONS+ETI(3)
      RETURN
      END


      SUBROUTINE INTX (EL1,EL2,B,IJ,SGR,SGI)
C
C     INTX PERFORMS NUMERICAL INTEGRATION OF EXP(JKR)/R BY THE METHOD OF
C     VARIABLE INTERVAL WIDTH ROMBERG INTEGRATION.  THE INTEGRAND VALUE
C     IS SUPPLIED BY SUBROUTINE GF.
C
      DATA NX,NM,NTS,RX/1,65536,4,1.E-4/
      Z=EL1
      ZE=EL2
      IF (IJ.EQ.0) ZE=0.
      S=ZE-Z
      FNM=NM
      EP=S/(10.*FNM)
      ZEND=ZE-EP
      SGR=0.
      SGI=0.
      NS=NX
      NT=0
      CALL GF (Z,G1R,G1I)
1     FNS=NS
      DZ=S/FNS
      DZOT=DZ*.5
      ZP=Z+DZ
      IF (ZP-ZE) 3,3,2
2     DZ=ZE-Z
      IF (ABS(DZ)-EP) 17,17,3
3     ZP=Z+DZOT
      CALL GF (ZP,G3R,G3I)
      ZP=Z+DZ
      CALL GF (ZP,G5R,G5I)
4     T00R=(G1R+G5R)*DZOT
      T00I=(G1I+G5I)*DZOT
      T01R=(T00R+DZ*G3R)*0.5
      T01I=(T00I+DZ*G3I)*0.5
      T10R=(4.0*T01R-T00R)/3.0
      T10I=(4.0*T01I-T00I)/3.0
C
C     TEST CONVERGENCE OF 3 POINT ROMBERG RESULT.
C
      CALL TEST (T01R,T10R,TE1R,T01I,T10I,TE1I)
      IF (TE1I-RX) 5,5,6
5     IF (TE1R-RX) 8,8,6
6     ZP=Z+DZ*0.25
      CALL GF (ZP,G2R,G2I)
      ZP=Z+DZ*0.75
      CALL GF (ZP,G4R,G4I)
      T02R=(T01R+DZOT*(G2R+G4R))*0.5
      T02I=(T01I+DZOT*(G2I+G4I))*0.5
      T11R=(4.0*T02R-T01R)/3.0
      T11I=(4.0*T02I-T01I)/3.0
      T20R=(16.0*T11R-T10R)/15.0
      T20I=(16.0*T11I-T10I)/15.0
C
C     TEST CONVERGENCE OF 5 POINT ROMBERG RESULT.
C
      CALL TEST (T11R,T20R,TE2R,T11I,T20I,TE2I)
      IF (TE2I-RX) 7,7,14
7     IF (TE2R-RX) 9,9,14
8     SGR=SGR+T10R
      SGI=SGI+T10I
      NT=NT+2
      GO TO 10
9     SGR=SGR+T20R
      SGI=SGI+T20I
      NT=NT+1
10    Z=Z+DZ
      IF (Z-ZEND) 11,17,17
11    G1R=G5R
      G1I=G5I
      IF (NT-NTS) 1,12,12
12    IF (NS-NX) 1,1,13
C
C     DOUBLE STEP SIZE
C
13    NS=NS/2
      NT=1
      GO TO 1
14    NT=0
      IF (NS-NM) 16,15,15
15    PRINT 20, Z
      GO TO 9
C
```

```
C     HALVE STEP SIZE
C
16    NS=NS*2
      FNS=NS
      DZ=S/FNS
      DZOT=DZ*0.5
      G5R=G3R
      G5I=G3I
      G3R=G2R
      G3I=G2I
      GO TO 4
17    CONTINUE
      IF (IJ) 19,18,19
C
C     ADD CONTRIBUTION OF NEAR SINGULARITY FOR DIAGONAL TERM
C
18    SGR=2.*(SGR+ALOG((SQRT(B*B+S*S)+S)/B))
      SGI=2.*SGI
19    CONTINUE
      RETURN
C
20    FORMAT (24H STEP SIZE LIMITED AT Z=,F10.5)
      END


      FUNCTION ISEGNO (ITAGI,MX)
C
C     ISEGNO RETURNS THE SEGMENT NUMBER OF THE MTH SEGMENT HAVING THE
C     TAG NUMBER ITAGI.  IF ITAGI=0 SEGMENT NUMBER M IS RETURNED.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      IF (MX.GT.0) GO TO 1
      PRINT 6
      STOP
1     ICNT=0
      IF (ITAGI.NE.0) GO TO 2
      ISEGNO=MX
      RETURN
2     IF (N.LT.1) GO TO 4
      DO 3 I=1,N
      IF (ITAG(I).NE.ITAGI) GO TO 3
      ICNT=ICNT+1
      IF (ICNT.EQ.MX) GO TO 5
3     CONTINUE
4     PRINT 7, ITAGI
      STOP
5     ISEGNO=I
      RETURN
C
6     FORMAT (4X,91HCHECK DATA, PARAMETER SPECIFYING SEGMENT POSITION IN
     1 A GROUP OF EQUAL TAGS MUST NOT BE ZERO)
7     FORMAT (///,10X,26HNO SEGMENT HAS AN ITAG OF ,I5)
      END


      SUBROUTINE JMELS (ETR,ETI,NCP,JP,NCM,JM,I,CM,NROW,NCOL)
C
C     JMELS SUMS THE CONTRIBUTIONS TO THE MATRIX ELEMENTS FOR SEGMENTS
C     CONNECTED TO JUNCTIONS OF THREE OR MORE SEGMENTS
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION CM(NROW,NCOL)
      DIMENSION JP(25), JM(25)
      COMPLEX CM
C     STATEMENT FUNCTION TO CALCULATE MATRIX LOCATION
      JL(K)=((K-1)/NP)*2*MP+K
      IF (NCP.LT.1) GO TO 2
      DO 1 J=1,NCP
      JPJ=JL(JP(J))
1     CM(JPJ,I)=CM(JPJ,I)+CMPLX(ETR,ETI)
2     CONTINUE
      IF (NCM.LT.1) GO TO 4
      DO 3 J=1,NCM
      JMJ=JL(JM(J))
3     CM(JMJ,I)=CM(JMJ,I)-CMPLX(ETR,ETI)
4     CONTINUE
      RETURN
      END


      SUBROUTINE JUNC (J,JNO,NC1,NSEG1,NC2,NSEG2,D)
C
C     JUNC SEARCHES CONNECTION DATA ARRAYS ICON1 AND ICON2 TO FIND ALL
C     CONNECTED SEGMENTS AT A MULTIPLE JUNCTION.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION NSEG1(25), NSEG2(25)
      NC1=0
      NC2=0
      SNC=0.0
```

```
      DO 4 I=1,N
      IF (ICON1(I)-JNO) 2,1,2
1     IF (I.EQ.J) GO TO 2
      NC1=NC1+1
      IF (NC1.GT.25) GO TO 5
      NSEG1(NC1)=I
      SNC=SNC+SI(I)
2     IF (ICON2(I)-JNO) 4,3,4
3     IF (I.EQ.J) GO TO 4
      NC2=NC2+1
      IF (NC2.GT.25) GO TO 5
      NSEG2(NC2)=I
      SNC=SNC+SI(I)
4     CONTINUE
      FC=NC1+NC2
      D=(SI(J)+SNC/FC)/2.0
      RETURN
5     PRINT 6, JNO
      STOP
C
6     FORMAT (41H ERROR - TOO MANY CONNECTIONS TO JUNCTION,I4)
      END


      SUBROUTINE LFACTR (A,NROW,NCOL,IX1,IX2,IP)
C
C     LFACTR PERFORMS GAUSS-DOOLITTLE MANIPULATIONS ON THE TWO BLOCKS OF
C     THE TRANSPOSED MATRIX IN CORE STORAGE.  THE GAUSS-DOOLITTLE
C     ALGORITHM IS PRESENTED ON PAGES 411-416 OF A. RALSTON -- A FIRST
C     COURSE IN NUMERICAL ANALYSIS.  COMMENTS BELOW REFER TO COMMENTS IN
C     RALSTONS TEXT.
C
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /SCRATM/ D(1500)
      DIMENSION A(NROW,NCOL), IP(NROW)
      COMPLEX A,D,AJR
      INTEGER R,R1,R2,PJ,PR
      LOGICAL L1,L2,L3
      IFLG=0
C
C     INITIALIZE R1,R2,J1,J2
C
      L1=IX1.EQ.1.AND.IX2.EQ.2
      L2=(IX2-1).EQ.IX1
      L3=IX2.EQ.NBLOKS
      IF (L1) GO TO 1
      GO TO 2
1     R1=1
      R2=2*NPBLK
      J1=1
      J2=-1
      GO TO 5
2     R1=NPBLK+1
      R2=2*NPBLK
      J1=(IX1-1)*NPBLK+1
      IF (L2) GO TO 3
      GO TO 4
3     J2=J1+NPBLK-2
      GO TO 5
4     J2=J1+NPBLK-1
5     IF (L3) R2=NPBLK+NLAST
      DO 16 R=R1,R2
C
C     STEP 1
C
      DO 6 K=J1,NROW
      D(K)=A(K,R)
6     CONTINUE
C
C     STEPS 2 AND 3
C
      IF (L1.OR.L2) J2=J2+1
      IF (J1.GT.J2) GO TO 9
      IXJ=0
      DO 8 J=J1,J2
      IXJ=IXJ+1
      PJ=IP(J)
      AJR=D(PJ)
      A(J,R)=AJR
      D(PJ)=D(J)
      JP1=J+1
      DO 7 I=JP1,NROW
      D(I)=D(I)-A(I,IXJ)*AJR
7     CONTINUE
8     CONTINUE
9     CONTINUE
C
C     STEP 4
C
      J2P1=J2+1
      IF (L1.OR.L2) GO TO 11
      IF (NROW.LT.J2P1) GO TO 16
      DO 10 I=J2P1,NROW
      A(I,R)=D(I)
10    CONTINUE
      GO TO 16
11    DMAX=REAL(D(J2P1)*CONJG(D(J2P1)))
      IP(J2P1)=J2P1
```

```
      J2P2=J2+2
      IF (J2P2.GT.NROW) GO TO 13
      DO 12 I=J2P2,NROW
      ELMAG=REAL(D(I)*CONJG(D(I)))
      IF (ELMAG.LT.DMAX) GO TO 12
      DMAX=ELMAG
      IP(J2P1)=I
12    CONTINUE
13    CONTINUE
      IF (DMAX.LT.1.E-10) IFLG=1
      PR=IP(J2P1)
      A(J2P1,R)=D(PR)
      D(PR)=D(J2P1)
C
C     STEP 5
C
      IF (J2P2.GT.NROW) GO TO 15
      AJR=1./A(J2P1,R)
      DO 14 I=J2P2,NROW
      A(I,R)=D(I)*AJR
14    CONTINUE
15    CONTINUE
      IF (IFLG.EQ.0) GO TO 16
      PRINT 17, J2,DMAX
      IFLG=0
16    CONTINUE
      RETURN
C
17    FORMAT (1H ,6HPIVOT(,I3,2H)=,E15.8)
      END


      SUBROUTINE LOAD (LDTYP,LDTAG,LDTAGF,LDTAGT,ZLR,ZLI,ZLC,NLOAD)
C
C     LOAD CALCULATES THE IMPEDANCE OF SPECIFIED SEGMENTS FOR VARIOUS
C     TYPES OF LOADING
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ZLOAD/ ZARRAY(1000)
      DIMENSION LDTYP(1), LDTAG(1), LDTAGF(1), LDTAGT(1), ZLR(1), ZLI(1)
     1, ZLC(1)
      COMPLEX ZARRAY,ZT,TPCJ,ZINT
      DATA TPCJ/(0.,1.88365371E+9)/
C
C     PRINT HEADING
C
      PRINT 23
C
C     INITIALIZE D ARRAY, USED FOR TEMPORARY STORAGE OF LOADING
C     INFORMATION.
C
      DO 1 I=1,N
1     ZARRAY(I)=(0.,0.)
      IWARN=0
C
C     CYCLE OVER LOADING CARDS
C
      ISTEP=0
2     ISTEP=ISTEP+1
      IF (ISTEP.LE.NLOAD) GO TO 3
      IF (IWARN.EQ.1) PRINT 24
      RETURN
3     IF (LDTYP(ISTEP).LE.5) GO TO 4
      PRINT 25, LDTYP(ISTEP)
      STOP
4     LDTAGS=LDTAG(ISTEP)
      JUMP=LDTYP(ISTEP)+1
      ICHK=0
C
C     SEARCH SEGMENTS FOR PROPER ITAGS
C
      L1=1
      L2=N
      IF (LDTAGS.NE.0) GO TO 5
      IF (LDTAGF(ISTEP).EQ.0.AND.LDTAGT(ISTEP).EQ.0) GO TO 5
      L1=LDTAGF(ISTEP)
      L2=LDTAGT(ISTEP)
5     DO 15 I=L1,L2
      IF (LDTAGS.EQ.0) GO TO 6
      IF (LDTAGS.NE.ITAG(I)) GO TO 15
      IF (LDTAGF(ISTEP).EQ.0) GO TO 6
      ICHK=ICHK+1
      IF (ICHK.GE.LDTAGF(ISTEP).AND.ICHK.LE.LDTAGT(ISTEP)) GO TO 7
      GO TO 15
6     ICHK=1
C
C     CALCULATION OF LAMDA*IMPED. PER UNIT LENGTH, JUMP TO APPROPRIATE
C     SECTION FOR LOADING TYPE
C
7     GO TO (8,9,10,11,12,13), JUMP
8     ZT=ZLR(ISTEP)/SI(I)+TPCJ*ZLI(ISTEP)/(SI(I)*WLAM)
      IF (ABS(ZLC(ISTEP)).GT.1.E-20) ZT=ZT+WLAM/(TPCJ*SI(I)*ZLC(ISTEP))
      GO TO 14
9     ZT=TPCJ*SI(I)*ZLC(ISTEP)/WLAM
      IF (ABS(ZLI(ISTEP)).GT.1.E-20) ZT=ZT+SI(I)*WLAM/(TPCJ*ZLI(ISTEP))
      IF (ABS(ZLR(ISTEP)).GT.1.E-20) ZT=ZT+SI(I)/ZLR(ISTEP)
```

```
      ZT=1./ZT
      GO TO 14
10    ZT=ZLR(ISTEP)*WLAM+TPCJ*ZLI(ISTEP)
      IF (ABS(ZLC(ISTEP)).GT.1.E-20) ZT=ZT+1./(TPCJ*SI(I)*SI(I)*ZLC(ISTE
     1P))
      GO TO 14
11    ZT=TPCJ*SI(I)*SI(I)*ZLC(ISTEP)
      IF (ABS(ZLI(ISTEP)).GT.1.E-20) ZT=ZT+1./(TPCJ*ZLI(ISTEP))
      IF (ABS(ZLR(ISTEP)).GT.1.E-20) ZT=ZT+1./(ZLR(ISTEP)*WLAM)
      ZT=1./ZT
      GO TO 14
12    ZT=CMPLX(ZLR(ISTEP),ZLI(ISTEP))/SI(I)
      GO TO 14
13    ZT=ZINT(ZLR(ISTEP)*WLAM,BI(I))
14    IF ((ABS(REAL(ZARRAY(I)))+ABS(AIMAG(ZARRAY(I)))).GT.1.E-20) IWARN=
     11
      ZARRAY(I)=ZARRAY(I)+ZT
15    CONTINUE
      IF (ICHK.NE.0) GO TO 16
      PRINT 26, LDTAGS
      STOP
C
C     PRINTING THE SEGMENT LOADING DATA, JUMP TO PROPER PRINT
C
16    GO TO (17,18,19,20,21,22), JUMP
17    CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP
     1),ZLC(ISTEP),0.,0.,0.,7H SERIES,7)
      GO TO 2
18    CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP
     1),ZLC(ISTEP),0.,0.,0.,8HPARALLEL,8)
      GO TO 2
19    CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP
     1),ZLC(ISTEP),0.,0.,0.,18HSERIES (PER METER),18)
      GO TO 2
20    CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP
     1),ZLC(ISTEP),0.,0.,0.,20HPARALLEL (PER METER),20)
      GO TO 2
21    CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),0.,0.,0.,ZLR(ISTEP),
     1ZLI(ISTEP),0.,15HFIXED IMPEDANCE,15)
      GO TO 2
22    CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),0.,0.,0.,0.,0.,ZLR(I
     1STEP),6H  WIRE,6)
      GO TO 2
C
23    FORMAT (//,7X,8HLOCATION,10X,10HRESISTANCE,3X,10HINDUCTANCE,2X,11H
     1CAPACITANCE,7X,16HIMPEDANCE (OHMS),5X,12HCONDUCTIVITY,4X,4HTYPE,/,
     24X,4HITAG,10H FROM THRU,10X,4HOHMS,8X,6HHENRYS,7X,6HFARADS,8X,4HRE
     3AL,6X,9HIMAGINARY,4X,10HMHOS/METER)
24    FORMAT (/,10X,74HNOTE, SOME OF THE ABOVE SEGMENTS HAVE BEEN LOADED
     1 TWICE - IMPEDANCES ADDED)
25    FORMAT (/,10X,46HIMPROPER LOAD TYPE CHOOSEN, REQUESTED TYPE IS ,I3
     1)
26    FORMAT (/,10X,50HLOADING DATA CARD ERROR, NO SEGMENT HAS AN ITAG =
     1 ,I5)
      END
```

```
      SUBROUTINE LTSOLV (A,NROW,NCOL,IX,B)
C
C     LTSOLV SOLVES THE MATRIX EQ. Y(R)*LU(T)=B(R) WHERE (R) DENOTES ROW
C     VECTOR AND LU(T) DENOTES THE LU DECOMPOSITION OF THE TRANSPOSE OF
C     THE ORIGINAL COEFFICIENT MATRIX.  THE LU(T) DECOMPOSITION IS
C     STORED ON TAPE 5 IN BLOCKS IN ASCENDING ORDER AND ON FILE 3 IN
C     BLOCKS OF DESCENDING ORDER.
C
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /SCRATM/ Y(1500)
      DIMENSION A(NROW,NCOL), B(NROW), IX(NROW)
      COMPLEX A,B,Y,SUM
C
C     FORWARD SUBSTITUTION
C
      I2=2*NPBLK*NROW
      J=0
      DO 4 IXBLK1=1,NBLOKS
      CALL BLCKIN (15,1,I2,1,121)
      K2=NPBLK
      IF (IXBLK1.EQ.NBLOKS) K2=NLAST
      DO 3 K=1,K2
      JM1=J
      J=J+1
      SUM=(0.,0.)
      IF (JM1.LT.1) GO TO 2
      DO 1 I=1,JM1
      SUM=SUM+A(I,K)*Y(I)
1     CONTINUE
2     CONTINUE
      Y(J)=(B(J)-SUM)/A(J,K)
3     CONTINUE
4     CONTINUE
C
C     BACKWARD SUBSTITUTION
C
      J=NROW+1
      DO 8 IXBLK1=1,NBLOKS
      CALL BLCKIN (16,1,I2,1,122)
      K2=NPBLK
      IF (IXBLK1.EQ.1) K2=NLAST
```

```
      DO 7 K=1,K2
      KP=K2-K+1
      JP1=J
      J=J-1
      SUM=(0.,0.)
      IF (NROW.LT.JP1) GO TO 6
      DO 5 I=JP1,NROW
      SUM=SUM+A(I,KP)*B(I)
5     CONTINUE
6     CONTINUE
      B(J)=Y(J)-SUM
7     CONTINUE
8     CONTINUE
C
C     UNSCRAMBLE SOLUTION
C
      DO 9 I=1,NROW
      IXI=IX(I)
      Y(IXI)=B(I)
9     CONTINUE
      DO 10 I=1,NROW
10    B(I)=Y(I)
      RETURN
      END


      SUBROUTINE LUNSCR (A,NROW,NCOL,IX,IP)
C
C     S/R WHICH UNSCRAMBLES, SCRAMBLED FACTORED MATRIX
C
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      DIMENSION A(NROW,NCOL), IP(NROW), IX(NROW)
      COMPLEX A,TEMP
      IF (IC3.EQ.-1) GO TO 9
      I1=1
      I2=2*NPBLK*NROW
      NM1=NROW-1
      DO 4 IXBLK1=1,NBLOKS
      CALL BLCKIN (12,I1,I2,1,121)
      K1=(IXBLK1-1)*NPBLK+2
      IF (NM1.LT.K1) GO TO 3
      J2=0
      DO 2 K=K1,NM1
      IF (J2.LT.NPBLK) J2=J2+1
      IPK=IP(K)
      DO 1 J=1,J2
      TEMP=A(K,J)
      A(K,J)=A(IPK,J)
      A(IPK,J)=TEMP
1     CONTINUE
2     CONTINUE
3     CONTINUE
      CALL BLCKOT (15,I1,I2,1,122)
4     CONTINUE
      DO 5 IXBLK1=1,NBLOKS
      BACKSPACE 15
      IF (IXBLK1.NE.1) BACKSPACE 15
      CALL BLCKIN (15,I1,I2,1,123)
      CALL BLCKOT (16,I1,I2,1,124)
5     CONTINUE
      DO 6 I=1,NROW
      IX(I)=I
6     CONTINUE
      DO 7 I=1,NROW
      IPI=IP(I)
      IXT=IX(I)
      IX(I)=IX(IPI)
      IX(IPI)=IXT
7     CONTINUE
      NB1=NBLOKS-1
      DO 8 IXBLK1=1,NB1
      CALL BLCKIN (15,I1,I2,1,125)
8     CONTINUE
C     SKIP NB1 LOGICAL RECORDS FORWARD
9     RETURN
      END


      SUBROUTINE MATFIL (ETR,ETI,IPR,J,JCO1,JCO2,CM,NROW,NCOL,IFLG)
C
C     MATFIL FILLS THE MATRIX ELEMENTS REPRESENTING FIELDS DUE TO WIRE
C     SEGMENT CURRENTS (EITHER E FIELD ON A SEGMENT OR H FIELD ON A
C     PATCH)
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /JUNK/ NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
      DIMENSION ETR(3), ETI(3), CM(NROW,NCOL)
      COMPLEX CM
C
C     FUNCTION TO CALCULATE MATRIX ELEMENT LOCATION
      JL(K)=((K-1)/NP)*2*MP+K
C
      IF (IFLG.EQ.0) GO TO 10
C
```

```
C     CONTRIBUTIONS TO ELEMENTS FOR SEGMENTS CONNECTED TO END ONE OF
C     SEGMENT J
C
      IF (JCO1) 1,5,2
1     CALL JMELS (ETR(1),ETI(1),NCIX,JIX,NCOX,JOX,IPR,CM,NROW,NCOL)
      GO TO 5
2     IF (JCO1.LT.10000) GO TO 3
      JLOC=JL(J)
      GO TO 4
3     JLOC=JL(JCO1)
      IF (ICON2(JCO1).EQ.J) GO TO 4
      IF (JCO1.EQ.J) GO TO 4
      CM(JLOC,IPR)=CM(JLOC,IPR)-CMPLX(ETR(1),ETI(1))
      GO TO 5
4     CM(JLOC,IPR)=CM(JLOC,IPR)+CMPLX(ETR(1),ETI(1))
C
C     CONTRIBUTIONS TO ELEMENTS FOR SEGMENTS CONNECTED TO END TWO OF
C     SEGMENT J
C
5     IF (JCO2) 6,10,7
6     CALL JMELS (ETR(3),ETI(3),NCOZ,JOZ,NCIZ,JIZ,IPR,CM,NROW,NCOL)
      GO TO 10
7     IF (JCO2.LT.10000) GO TO 8
      JLOC=JL(J)
      GO TO 9
8     JLOC=JL(JCO2)
      IF (ICON1(JCO2).EQ.J) GO TO 9
      IF (JCO2.EQ.J) GO TO 9
      CM(JLOC,IPR)=CM(JLOC,IPR)-CMPLX(ETR(3),ETI(3))
      GO TO 10
9     CM(JLOC,IPR)=CM(JLOC,IPR)+CMPLX(ETR(3),ETI(3))
C
C     CONTRIBUTION TO ELEMENT FOR SEGMENT J
C
10    CONTINUE
      JLOC=JL(J)
      CM(JLOC,IPR)=CM(JLOC,IPR)+CMPLX(ETR(2),ETI(2))
      RETURN
      END
```

```
      SUBROUTINE MOVE (ROX,ROY,ROZ,XS,YS,ZS,ITS,NRPT,ITGI)
C
C     SUBROUTINE MOVE MOVES THE STRUCTURE WITH RESPECT TO ITS
C     COORDINATE SYSTEM OR REPRODUCES STRUCTURE IN NEW POSITIONS.
C     STRUCTURE IS ROTATED ABOUT X,Y,Z AXES BY ROX,ROY,ROZ
C     RESPECTIVELY, THEN SHIFTED BY XS,YS,ZS
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), X2(1), Y
     12(1), Z2(1)
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      IF (ABS(ROX)+ABS(ROY).GT.1.E-10) IPSYM=IPSYM*3
      SPS=SIN(ROX)
      CPS=COS(ROX)
      STH=SIN(ROY)
      CTH=COS(ROY)
      SPH=SIN(ROZ)
      CPH=COS(ROZ)
      XX=CPH*CTH
      XY=CPH*STH*SPS-SPH*CPS
      XZ=CPH*STH*CPS+SPH*SPS
      YX=SPH*CTH
      YY=SPH*STH*SPS+CPH*CPS
      YZ=SPH*STH*CPS-CPH*SPS
      ZX=-STH
      ZY=CTH*SPS
      ZZ=CTH*CPS
      NRP=NRPT
      IF (NRPT.EQ.0) NRP=1
C
C     TRANSFORM COORDINATES OF SEGMENTS
C
      IF (N.EQ.0) GO TO 3
      I1=ISEGNO(ITS,1)
      IX=I1
      K=N
      IF (NRPT.EQ.0) K=I1-1
      DO 2 IR=1,NRP
      DO 1 I=I1,N
      K=K+1
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)
      X(K)=XI*XX+YI*XY+ZI*XZ+XS
      Y(K)=XI*YX+YI*YY+ZI*YZ+YS
      Z(K)=XI*ZX+YI*ZY+ZI*ZZ+ZS
      XI=X2(I)
      YI=Y2(I)
      ZI=Z2(I)
      X2(K)=XI*XX+YI*XY+ZI*XZ+XS
      Y2(K)=XI*YX+YI*YY+ZI*YZ+YS
      Z2(K)=XI*ZX+YI*ZY+ZI*ZZ+ZS
```

```
      BI(K)=BI(I)
      ITAG(K)=ITAG(I)+ITGI
1     CONTINUE
      I1=N+1
      N=K
2     CONTINUE
C
C     TRANSFORM COORDINATES OF PATCHES
C
3     IF (M.EQ.0) GO TO 6
      I1=1
      K=M
      LD1=LD+1
      IF (NRPT.EQ.0) K=0
      DO 5 I1=1,NRP
      DO 4 I=I1,M
      K=K+1
      IR=LD1-I
      KR=LD1-K
      XI=X(IR)
      YI=Y(IR)
      ZI=Z(IR)
      X(KR)=XI*XX+YI*XY+ZI*XZ+XS
      Y(KR)=XI*YX+YI*YY+ZI*YZ+YS
      Z(KR)=XI*ZX+YI*ZY+ZI*ZZ+ZS
      XI=T1X(IR)
      YI=T1Y(IR)
      ZI=T1Z(IR)
      T1X(KR)=XI*XX+YI*XY+ZI*XZ
      T1Y(KR)=XI*YX+YI*YY+ZI*YZ
      T1Z(KR)=XI*ZX+YI*ZY+ZI*ZZ
      XI=T2X(IR)
      YI=T2Y(IR)
      ZI=T2Z(IR)
      T2X(KR)=XI*XX+YI*XY+ZI*XZ
      T2Y(KR)=XI*YX+YI*YY+ZI*YZ
      T2Z(KR)=XI*ZX+YI*ZY+ZI*ZZ
      SALP(KR)=SALP(IR)
4     BI(KR)=BI(IR)
      I1=M+1
5     M=K
6     IF ((NRPT.EQ.0).AND.(IX.EQ.1)) RETURN
      NP=N
      MP=M
      IPSYM=0
      RETURN
      END


      SUBROUTINE NEFLD (XOB,YOB,ZOB,EX,EY,EZ)
C
C     NEFLD COMPUTES THE NEAR FIELD AT SPECIFIED POINTS IN SPACE AFTER
C     THE STRUCTURE CURRENTS HAVE BEEN COMPUTED.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      COMMON /CRNT/ AIR(1000),AII(1000),BIR(1000),BII(1000),CIR(1000),CI
     1I(1000),CUR(2000)
      COMMON /REFL/ QX,QY,QZ,CABI,SABI,SALPI,PX,PY,REFS,REFPS
      COMMON /GND/ ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,IPERF
      DIMENSION CAB(1), SAB(1)
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1))
      COMPLEX FJ,EZP,ERHO,EX,EY,EZ,CUR,ZRSIN,REFS,REFPS,ZRATI,ZRATI2
      DATA FJ/(0.,1.)/
      EX=(0.,0.)
      EY=(0.,0.)
      EZ=(0.,0.)
C
C     SUMMATION OVER FIELDS OF INDIVIDUAL SEGMENTS.
C
      DO 7 I=1,N
      CABI=CAB(I)
      SABI=SAB(I)
      SALPI=-SALP(I)
      XD=XOB-X(I)
      YD=YOB-Y(I)
      RFL=-1.
C
C     LOOP TO INCLUDE FIELD OF IMAGE OF SEGMENT WHEN GROUND IS SPECIFIED
C
      DO 7 J=1,KSYMP
      RFL=-RFL
      ZD=ZOB-Z(I)*RFL
      SALPI=-SALPI
      ZP=XD*CABI+YD*SABI+ZD*SALPI
      XYMAG=XD*XD+YD*YD
      RS=XYMAG+ZD*ZD
      RH2=RS-ZP*ZP
      IF (RH2.LT.1.E-20) GO TO 1
      RH=SQRT(RH2)
      QX=(XD-ZP*CABI)/RH
      QY=(YD-ZP*SABI)/RH
      QZ=(ZD-ZP*SALPI)/RH
      GO TO 2
1     RH=0.
      QX=0.
```

```
      QY=0.
      QZ=0.
2     CALL EFLD (BI(I),SI(I),RH,ZP,1,EZRS,EZIS,ERRS,ERIS,EZRC,EZIC,ERRC,
     1ERIC,EZRK,EZIK,ERRK,ERIK)
      IF (J.NE.2) GO TO 6
      IF (IPERF.EQ.1) GO TO 5
      RMAG=SQRT(RS)
      XYMAG=SQRT(XYMAG)
      IF (XYMAG.GT.1.E-6) GO TO 3
      PX=0.
      PY=0.
      CTH=1.
      ZRSIN=(1.,0.)
      GO TO 4
3     PX=-YD/XYMAG
      PY=XD/XYMAG
      CTH=ZD/RMAG
      ZRSIN=CSQRT(1.-ZRATI*ZRATI*(1.-CTH*CTH))
4     REFS=-(CTH-ZRATI*ZRSIN)/(CTH+ZRATI*ZRSIN)
      REFPS=(ZRATI*CTH-ZRSIN)/(ZRATI*CTH+ZRSIN)
      REFPS=REFPS-REFS
5     CALL GN (EZRS,EZIS,ERRS,ERIS)
      CALL GN (EZRC,EZIC,ERRC,ERIC)
      CALL GN (EZRK,EZIK,ERRK,ERIK)
6     CONTINUE
      EZP=EZRK*AIR(I)-EZIK*AII(I)+EZRS*BIR(I)-EZIS*BII(I)+EZRC*CIR(I)-EZ
     1IC*CII(I)+FJ*(EZRK*AII(I)+EZIK*AIR(I)+EZRS*BII(I)+EZIS*BIR(I)+EZRC
     2*CII(I)+EZIC*CIR(I))
      ERHO=ERRK*AIR(I)-ERIK*AII(I)+ERRS*BIR(I)-ERIS*BII(I)+ERRC*CIR(I)-E
     1RIC*CII(I)+FJ*(ERRK*AII(I)+ERIK*AIR(I)+ERRS*BII(I)+ERIS*BIR(I)+ERR
     2C*CII(I)+ERIC*CIR(I))
      EX=EX+EZP*CABI+ERHO*QX
      EY=EY+EZP*SABI+ERHO*QY
      EZ=EZ+EZP*SALPI+ERHO*QZ
7     CONTINUE
      RETURN
      END


      SUBROUTINE NETWK (ISEG1,ISEG2,X11R,X11I,X12R,X12I,X22R,X22I,NONET,
     1NTYP,ISANT,VSANT,NSANT,CM,IP,EINC,NROW,NCOL,IX,PIN,PNLS,NPRINT,MAS
     2YM,ZPED,NTSOL)
C
C     SUBROUTINE NETWK SOLVES FOR STRUCTURE CURRENTS FOR A GIVEN
C     EXCITATION INCLUDING THE EFFECT OF NON-RADIATING NETWORKS IF
C     PRESENT.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION ISEG1(30), ISEG2(30), X11R(30), X11I(30), X12R(30), X12I
     1(30), X22R(30), X22I(30), ISANT(10), VSANT(10), NTYP(30)
      DIMENSION CM(NROW,NCOL), EINC(NROW), IP(NROW), IX(NROW)
      DIMENSION CMN(30,30), RHNT(30), IPNT(30), NTEQA(30), NTSCA(30), RH
     1S(1500), VSRC(10), RHNX(30)
      COMPLEX CMN,RHNT,YMIT,RHS,ZPED,CM,EINC,VSANT,VLT,CUR,VSRC,RHNX
      DATA NDIMN,NDIMNP/30,31/,TP/6.283185308/
      NEQ=N+2*M
      PIN=0.
      PNLS=0.
      IF (NTSOL.NE.0) GO TO 42
      NOP=NEQ/(NP+2*MP)
      IF (MASYM.EQ.0) GO TO 14
C
C     COMPUTE RELATIVE MATRIX ASYMMETRY
C
      IROW1=0
      IF (NONET.EQ.0) GO TO 5
      DO 4 I=1,NONET
      NSEG1=ISEG1(I)
      DO 3 ISC1=1,2
      IF (IROW1.EQ.0) GO TO 2
      DO 1 J=1,IROW1
      IF (NSEG1.EQ.IPNT(J)) GO TO 3
1     CONTINUE
2     IROW1=IROW1+1
      IPNT(IROW1)=NSEG1
3     NSEG1=ISEG2(I)
4     CONTINUE
5     IF (NSANT.EQ.0) GO TO 9
      DO 8 I=1,NSANT
      NSEG1=ISANT(I)
      IF (IROW1.EQ.0) GO TO 7
      DO 6 J=1,IROW1
      IF (NSEG1.EQ.IPNT(J)) GO TO 8
6     CONTINUE
7     IROW1=IROW1+1
      IPNT(IROW1)=NSEG1
8     CONTINUE
9     IF (IROW1.LT.NDIMNP) GO TO 10
      PRINT 57
      STOP
10    IF (IROW1.LT.2) GO TO 14
      DO 12 I=1,IROW1
      ISC1=IPNT(I)
      ASM=SI(ISC1)
      DO 11 J=1,NEQ
11    RHS(J)=(0.,0.)
```

```
      RHS(ISC1)=(1.,0.)
      CALL SOLVES (NOP,CM,IP,RHS,NROW,NCOL,IA,NP,N,MP,M)
      DO 12 J=1,IROW1
      ISC1=IPNT(J)
12    CMN(J,I)=RHS(ISC1)/ASM
      ASM=0.
      ASA=0.
      DO 13 I=2,IROW1
      ISC1=I-1
      DO 13 J=1,ISC1
      CUR=CMN(I,J)
      PWR=CABS((CUR-CMN(J,I))/CUR)
      ASA=ASA+PWR*PWR
      IF (PWR.LT.ASM) GO TO 13
      ASM=PWR
      NTEQ=IPNT(I)
      NTSC=IPNT(J)
13    CONTINUE
      ASA=SQRT(ASA*2./FLOAT(IROW1*(IROW1-1)))
      PRINT 56, ASM,NTEQ,NTSC,ASA
14    IF (NONET.EQ.0) GO TO 48
C
C     SOLUTION OF NETWORK EQUATIONS
C
      DO 15 I=1,NDIMN
      RHNX(I)=(0.,0.)
      DO 15 J=1,NDIMN
15    CMN(I,J)=(0.,0.)
      NTEQ=0
      NTSC=0
C
C     SORT NETWORK AND SOURCE DATA AND ASSIGN EQUATION NUMBERS TO
C     SEGMENTS.
C
      DO 38 J=1,NONET
      NSEG1=ISEG1(J)
      NSEG2=ISEG2(J)
      IF (NTYP(J).GT.1) GO TO 16
      Y11R=X11R(J)
      Y11I=X11I(J)
      Y12R=X12R(J)
      Y12I=X12I(J)
      Y22R=X22R(J)
      Y22I=X22I(J)
      GO TO 17
16    Y22R=TP*X11I(J)/WLAM
      Y12R=0.
      Y12I=1./(X11R(J)*SIN(Y22R))
      Y11R=X12R(J)
      Y11I=-Y12I*COS(Y22R)
      Y22R=X22R(J)
      Y22I=Y11I+X22I(J)
      Y11I=Y11I+X12I(J)
      IF (NTYP(J).EQ.2) GO TO 17
      Y12R=-Y12R
      Y12I=-Y12I
17    IF (NSANT.EQ.0) GO TO 19
      DO 18 I=1,NSANT
      IF (NSEG1.NE.ISANT(I)) GO TO 18
      ISC1=I
      GO TO 22
18    CONTINUE
19    ISC1=0
      IF (NTEQ.EQ.0) GO TO 21
      DO 20 I=1,NTEQ
      IF (NSEG1.NE.NTEQA(I)) GO TO 20
      IROW1=I
      GO TO 25
20    CONTINUE
21    NTEQ=NTEQ+1
      IROW1=NTEQ
      NTEQA(NTEQ)=NSEG1
      GO TO 25
22    IF (NTSC.EQ.0) GO TO 24
      DO 23 I=1,NTSC
      IF (NSEG1.NE.NTSCA(I)) GO TO 23
      IROW1=NDIMNP-I
      GO TO 25
23    CONTINUE
24    NTSC=NTSC+1
      IROW1=NDIMNP-NTSC
      NTSCA(NTSC)=NSEG1
      VSRC(NTSC)=VSANT(ISC1)
25    IF (NSANT.EQ.0) GO TO 27
      DO 26 I=1,NSANT
      IF (NSEG2.NE.ISANT(I)) GO TO 26
      ISC2=I
      GO TO 30
26    CONTINUE
27    ISC2=0
      IF (NTEQ.EQ.0) GO TO 29
      DO 28 I=1,NTEQ
      IF (NSEG2.NE.NTEQA(I)) GO TO 28
      IROW2=I
      GO TO 33
28    CONTINUE
29    NTEQ=NTEQ+1
      IROW2=NTEQ
      NTEQA(NTEQ)=NSEG2
```

```
      GO TO 33
30    IF (NTSC.EQ.0) GO TO 32
      DO 31 I=1,NTSC
      IF (NSEG2.NE.NTSCA(I)) GO TO 31
      IROW2=NDIMNP-I
      GO TO 33
31    CONTINUE
32    NTSC=NTSC+1
      IROW2=NDIMNP-NTSC
      NTSCA(NTSC)=NSEG2
      VSRC(NTSC)=VSANT(ISC2)
33    IF (NTSC+NTEQ.LT.NDIMNP) GO TO 34
      PRINT 57
      STOP
C
C     FILL NETWORK EQUATION MATRIX AND RIGHT HAND SIDE VECTOR WITH
C     NETWORK SHORT-CIRCUIT ADMITTANCE MATRIX COEFFICIENTS.
C
34    IF (ISC1.NE.0) GO TO 35
      CMN(IROW1,IROW1)=CMN(IROW1,IROW1)-CMPLX(Y11R,Y11I)*SI(NSEG1)
      CMN(IROW1,IROW2)=CMN(IROW1,IROW2)-CMPLX(Y12R,Y12I)*SI(NSEG1)
      GO TO 36
35    RHNX(IROW1)=RHNX(IROW1)+CMPLX(Y11R,Y11I)*VSANT(ISC1)/WLAM
      RHNX(IROW2)=RHNX(IROW2)+CMPLX(Y12R,Y12I)*VSANT(ISC1)/WLAM
36    IF (ISC2.NE.0) GO TO 37
      CMN(IROW2,IROW2)=CMN(IROW2,IROW2)-CMPLX(Y22R,Y22I)*SI(NSEG2)
      CMN(IROW2,IROW1)=CMN(IROW2,IROW1)-CMPLX(Y12R,Y12I)*SI(NSEG2)
      GO TO 38
37    RHNX(IROW1)=RHNX(IROW1)+CMPLX(Y12R,Y12I)*VSANT(ISC2)/WLAM
      RHNX(IROW2)=RHNX(IROW2)+CMPLX(Y22R,Y22I)*VSANT(ISC2)/WLAM
38    CONTINUE
C
C     ADD INTERACTION MATRIX ADMITTANCE ELEMENTS TO NETWORK EQUATION
C     MATRIX
C
      DO 41 I=1,NTEQ
      DO 39 J=1,NEQ
39    RHS(J)=(0.,0.)
      IROW1=NTEQA(I)
      RHS(IROW1)=(1.,0.)
      CALL SOLVES (NOP,CM,IP,RHS,NROW,NCOL,IX,NP,N,MP,M)
      DO 40 J=1,NTEQ
      IROW1=NTEQA(J)
40    CMN(I,J)=CMN(I,J)+RHS(IROW1)
41    CONTINUE
C
C     FACTOR NETWORK EQUATION MATRIX
C
      CALL FACTR (NTEQ,CMN,IPNT,NDIMN)
C
C     ADD TO NETWORK EQUATION RIGHT HAND SIDE THE TERMS DUE TO ELEMENT
C     INTERACTIONS
C
42    IF (NONET.EQ.0) GO TO 48
      DO 43 I=1,NEQ
43    RHS(I)=EINC(I)
      CALL SOLVES (NOP,CM,IP,RHS,NROW,NCOL,IX,NP,N,MP,M)
      DO 44 I=1,NTEQ
      IROW1=NTEQA(I)
44    RHNT(I)=RHNX(I)+RHS(IROW1)
C
C     SOLVE NETWORK EQUATIONS
C
      CALL SOLVE (NTEQ,CMN,IPNT,RHNT,NDIMN)
C
C     ADD FIELDS DUE TO NETWORK VOLTAGES TO ELECTRIC FIELDS APPLIED TO
C     STRUCTURE AND SOLVE FOR INDUCED CURRENT
C
      DO 45 I=1,NTEQ
      IROW1=NTEQA(I)
45    EINC(IROW1)=EINC(IROW1)-RHNT(I)
      CALL SOLVES (NOP,CM,IP,EINC,NROW,NCOL,IX,NP,N,MP,M)
      IF (NPRINT.EQ.0) PRINT 59
      IF (NPRINT.EQ.0) PRINT 58
      DO 46 I=1,NTEQ
      IROW1=NTEQA(I)
      VLT=RHNT(I)*SI(IROW1)*WLAM
      CUR=EINC(IROW1)*WLAM
      YMIT=CUR/VLT
      ZPED=VLT/CUR
      IROW2=ITAG(IROW1)
      PWR=.5*REAL(VLT*CONJG(CUR))
      PNLS=PNLS-PWR
46    IF (NPRINT.EQ.0) PRINT 60, IROW2,IROW1,VLT,CUR,ZPED,YMIT,PWR
      IF (NTSC.EQ.0) GO TO 49
      DO 47 I=1,NTSC
      IROW1=NTSCA(I)
      VLT=VSRC(I)
      CUR=EINC(IROW1)*WLAM
      YMIT=CUR/VLT
      ZPED=VLT/CUR
      IROW2=ITAG(IROW1)
      PWR=.5*REAL(VLT*CONJG(CUR))
      PNLS=PNLS-PWR
47    IF (NPRINT.EQ.0) PRINT 60, IROW2,IROW1,VLT,CUR,ZPED,YMIT,PWR
      GO TO 49
C
C     SOLVE FOR CURRENTS WHEN NO NETWORKS ARE PRESENT
C
```

```
48    CALL SOLVES (NOP,CM,IP,EINC,NROW,NCOL,IX,NP,N,MP,M)
      NTSC=0
49    IF (NSANT.EQ.0) RETURN
      PRINT 61
      PRINT 58
      DO 55 I=1,NSANT
      ISC1=ISANT(I)
      VLT=VSANT(I)
      IF (NTSC.EQ.0) GO TO 51
      DO 50 J=1,NTSC
      IF (NTSCA(J).EQ.ISC1) GO TO 52
50    CONTINUE
51    CUR=EINC(ISC1)*WLAM
      IROW1=0
      GO TO 54
52    IROW1=NDIMNP-J
      CUR=RHNX(IROW1)
      DO 53 J=1,NTEQ
53    CUR=CUR-CMN(J,IROW1)*RHNT(J)
      CUR=(EINC(ISC1)+CUR)*WLAM
54    YMIT=CUR/VLT
      ZPED=VLT/CUR
      PWR=.5*REAL(VLT*CONJG(CUR))
      PIN=PIN+PWR
      IF (IROW1.NE.0) PNLS=PNLS+PWR
      IROW2=ITAG(ISC1)
55    PRINT 60, IROW2,ISC1,VLT,CUR,ZPED,YMIT,PWR
      RETURN
C
56    FORMAT (  ///,3X,47HMAXIMUM RELATIVE ASYMMETRY OF THE DRIVING POIN
     1T,21H ADMITTANCE MATRIX IS,E10.3,13H FOR SEGMENTS,I5,4H AND,I5,/,
     2  3X,25HRMS RELATIVE ASYMMETRY IS,F10.3)
57    FORMAT (1X,44HERROR - - NETWORK ARRAY DIMENSIONS TOO SMALL)
58    FORMAT (3X,3HTAG,3X,4HSEG.,5X,15HVOLTAGE (VOLTS),11X,14HCURRENT (A
     1MPS),12X,16HIMPEDANCE (OHMS),10X,17HADMITTANCE (MHOS),8X,5HPOWER,/
     2,3X,3HNO.,3X,3HNO.,5X,4HREAL,9X,5HIMAG.,3(8X,4HREAL,9X,5HIMAG.),7X
     3,7H(WATTS))
59    FORMAT (///,32X,66H- - - STRUCTURE EXCITATION DATA AT NETWORK CONN
     1ECTION POINTS - - -,/)
60    FORMAT (2(1X,I5),9E13.5)
61    FORMAT (///,46X,36H- - - ANTENNA INPUT PARAMETERS - - -,/)
      END


      SUBROUTINE PATCH (NX,NY,XC,YC,ZC,AL,BT,AR)
C
C     PATCH PERFORMS VARIOUS OPERATIONS ON THE GEOMETRY DATA FOR PATCHES
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
C
C     SET PARAMETERS TO DEFINE A NEW PATCH
C
      M=M+1
      IPSYM=0
      NP=N
      MP=M
      MI=LD+1-M
      X(MI)=XC
      Y(MI)=YC
      Z(MI)=ZC
      BI(MI)=AR
      ZW2=COS(AL)
      XW2=ZW2*COS(BT)
      YW2=ZW2*SIN(BT)
      ZW2=SIN(AL)
      XA=SQRT(XW2*XW2+YW2*YW2)
      IF (XA.LT.1.E-6) GO TO 1
      T1X(MI)=-YW2/XA
      T1Y(MI)=XW2/XA
      T1Z(MI)=0.
      GO TO 2
1     T1X(MI)=1.
      T1Y(MI)=0.
      T1Z(MI)=0.
2     T2X(MI)=YW2*T1Z(MI)-ZW2*T1Y(MI)
      T2Y(MI)=ZW2*T1X(MI)-XW2*T1Z(MI)
      T2Z(MI)=XW2*T1Y(MI)-YW2*T1X(MI)
      SALP(MI)=1.
      RETURN
C
C     GENERATE SURFACE BY MULTIPE SHIFTING OF THE LAST PATCH INPUT.
C
      ENTRY PACHS
      MI=LD+1-M
      XST=X(MI)-XC
      YW1=Y(MI)-YC
      ZW1=Z(MI)
      S1X=T1X(MI)
      S1Y=T1Y(MI)
      S1Z=T1Z(MI)
      S2X=T2X(MI)
      S2Y=T2Y(MI)
```

```
      S2Z=T2Z(MI)
      XA=BI(MI)
      SALN=SALP(MI)
      NXP=NX+1
      NYP=NY+1
C     LOOP FOR SHIFTING PATCH IN Y
      DO 4 IY=1,NYP
      YW1=YW1+YC
      XW1=XST
C     LOOP FOR SHIFTING PATCH IN X
      DO 3 IX=1,NXP
      XW1=XW1+XC
      IF (IX.EQ.1.AND.IY.EQ.1) GO TO 3
      M=M+1
      MI=MI-1
      X(MI)=XW1
      Y(MI)=YW1
      Z(MI)=ZW1
      BI(MI)=XA
      SALP(MI)=SALN
      T1X(MI)=S1X
      T1Y(MI)=S1Y
      T1Z(MI)=S1Z
      T2X(MI)=S2X
      T2Y(MI)=S2Y
      T2Z(MI)=S2Z
3     CONTINUE
4     CONTINUE
      NP=N
      MP=M
      IPSYM=0
      RETURN
C
C     SUBDIVIDE PATCH TO WHICH A SEGMENT END IS CONNECTED
C
      ENTRY SUBPH
      IF (NX.EQ.M) GO TO 6
C     SHIFT DATA FOR PATCHES  NX+1  THROUGH  M  DOWN BY 3 LOCATIONS IN
C     ARRAYS
      NXP=NX+1
      IX=LD-M
      DO 5 IY=NXP,M
      IX=IX+1
      NYP=IX-3
      X(NYP)=X(IX)
      Y(NYP)=Y(IX)
      Z(NYP)=Z(IX)
      BI(NYP)=BI(IX)
      SALP(NYP)=SALP(IX)
      T1X(NYP)=T1X(IX)
      T1Y(NYP)=T1Y(IX)
      T1Z(NYP)=T1Z(IX)
      T2X(NYP)=T2X(IX)
      T2Y(NYP)=T2Y(IX)
5     T2Z(NYP)=T2Z(IX)
C     INSERT DATA FOR SUBDIVIDED PATCHES INTO ARRAYS
6     MI=LD+1-NX
      XW1=X(MI)
      YW1=Y(MI)
      ZW1=Z(MI)
      XA=BI(MI)*.25
      XST=SQRT(XA)*.5
      S1X=T1X(MI)
      S1Y=T1Y(MI)
      S1Z=T1Z(MI)
      S2X=T2X(MI)
      S2Y=T2Y(MI)
      S2Z=T2Z(MI)
      SALN=SALP(MI)
      XW2=XST
      YW2=XST
      DO 9 IX=1,4
      X(MI)=XW1+XW2*S1X+YW2*S2X
      Y(MI)=YW1+XW2*S1Y+YW2*S2Y
      Z(MI)=ZW1+XW2*S1Z+YW2*S2Z
      BI(MI)=XA
      IF (IX.NE.1) GO TO 7
      XW2=-XW2
      GO TO 8
7     T1X(MI)=S1X
      T1Y(MI)=S1Y
      T1Z(MI)=S1Z
      T2X(MI)=S2X
      T2Y(MI)=S2Y
      T2Z(MI)=S2Z
      SALP(MI)=SALN
      IF (IX.EQ.2) YW2=-YW2
      IF (IX.EQ.3) XW2=-XW2
8     MI=MI-1
9     CONTINUE
      M=M+3
      IF (NX.LE.MP) MP=MP+3
      RETURN
      END


      SUBROUTINE PCINT (X1,Y1,Z1,XA,CABI,SABI,SALPI,E)
C
```

```
C     PCINT PERFORMS THE NUMERICAL INTEGRATION OF THE SURFACE CURRENT
C     INTERPOLATION FUNCTION OVER THE FOUR PATCHES AT THE BASE OF A
C     CONNECTED WIRE TO YIELD THE E FIELD ON THE CONNECTING WIRE SEGMENT
C
      COMMON /DATI/ T1XI,T1YI,T1ZI,T2XI,T2YI,T2ZI,XI,YI,ZI
      DIMENSION E(9)
      COMPLEX E1,E2,E3,E4,E5,E6,E7,E8,E9,E1X,E1Y,E1Z,E2X,E2Y,E2Z,E
      DATA TPI/6.283185308/,NINT/10/
      D=SQRT(XA)*.5
      DS=4.*D/FLOAT(NINT)
      DA=DS*DS
      GCON=1./XA
      FCON=1./(2.*TPI*D)
      S1=D+DS*.5
      XSS=XI+S1*(T1XI+T2XI)
      YSS=YI+S1*(T1YI+T2YI)
      ZSS=ZI+S1*(T1ZI+T2ZI)
      S1=S1+D
      S2X=S1
      E1=(0.,0.)
      E2=(0.,0.)
      E3=(0.,0.)
      E4=(0.,0.)
      E5=(0.,0.)
      E6=(0.,0.)
      E7=(0.,0.)
      E8=(0.,0.)
      E9=(0.,0.)
C     INTEGRATION OVER THE 4 PATCHES WITH  NINT X NINT  STEPS
      DO 1 I1=1,NINT
      S1=S1-DS
      S2=S2X
      XSS=XSS-DS*T1XI
      YSS=YSS-DS*T1YI
      ZSS=ZSS-DS*T1ZI
      XS=XSS
      YS=YSS
      ZS=ZSS
      DO 1 I2=1,NINT
      S2=S2-DS
      XS=XS-DS*T2XI
      YS=YS-DS*T2YI
      ZS=ZS-DS*T2ZI
      CALL UNERE (XI-XS,YI-YS,ZI-ZS,DA,E1X,E1Y,E1Z,E2X,E2Y,E2Z)
      E1X=E1X*CABI+E1Y*SABI+E1Z*SALPI
      E2X=E2X*CABI+E2Y*SABI+E2Z*SALPI
      G1=(D+S1)*(D+S2)*GCON
      G2=(D-S1)*(D+S2)*GCON
      G3=(D-S1)*(D-S2)*GCON
      G4=(D+S1)*(D-S2)*GCON
      F2=(S1*S1+S2*S2)*TPI
      F1=S1/F2-(G1-G2-G3+G4)*FCON
      F2=S2/F2-(G1+G2-G3-G4)*FCON
C     SUMMATION OF THE 9 COMPONENTS - TWO SURFACE COMPONENTS EACH FOR
C     THE FOUR PATCH CURRENTS PLUS THE SEGMENT CURRENT
      E1=E1+E1X*G1
      E2=E2+E1X*G2
      E3=E3+E1X*G3
      E4=E4+E1X*G4
      E5=E5+E2X*G1
      E6=E6+E2X*G2
      E7=E7+E2X*G3
      E8=E8+E2X*G4
1     E9=E9+E1X*F1+E2X*F2
      E(1)=E1
      E(2)=E2
      E(3)=E3
      E(4)=E4
      E(5)=E5
      E(6)=E6
      E(7)=E7
      E(8)=E8
      E(9)=E9
      RETURN
      END


      SUBROUTINE PRNT (IN1,IN2,IN3,FL1,FL2,FL3,FL4,FL5,FL6,IA,ICHAR)
C
C     PRNT SETS UP THE PRINT FORMATS FOR IMPEDANCE LOADING
C
      DIMENSION IVAR(13), IA(1), IFORM(8), IN(3), INT(3), FL(6), FLT(6)
      INTEGER HALL
      DATA IFORM/5H(/3X,,3HI5,,3H5X,,3HA5,,6HE13.4,,4H13X,,3H3X,,5H2A10)
     1/
C
C     NUMBER OF CHARACTERS PER COMPUTER WORD IS NCPW
C
      DATA NCPW/10/,HALL/4H ALL/
      NWORDS=(ICHAR-1)/NCPW+1
      IN(1)=IN1
      IN(2)=IN2
      IN(3)=IN3
      FL(1)=FL1
      FL(2)=FL2
      FL(3)=FL3
      FL(4)=FL4
      FL(5)=FL5
```

```
      FL(6)=FL6
C
C     INTEGER FORMAT
C
      NINT=0
      IVAR(1)=IFORM(1)
      K=1
      I1=1
      IF (.NOT.(IN1.EQ.0.AND.IN2.EQ.0.AND.IN3.EQ.0)) GO TO 1
      INT(1)=HALL
      NINT=1
      I1=2
      K=K+1
      IVAR(K)=IFORM(4)
1     DO 3 I=I1,3
      K=K+1
      IF (IN(I).EQ.0) GO TO 2
      NINT=NINT+1
      INT(NINT)=IN(I)
      IVAR(K)=IFORM(2)
      GO TO 3
2     IVAR(K)=IFORM(3)
3     CONTINUE
      K=K+1
      IVAR(K)=IFORM(7)
C
C     FLOATING POINT FORMAT
C
      NFLT=0
      DO 5 I=1,6
      K=K+1
      IF (ABS(FL(I)).LT.1.E-20) GO TO 4
      NFLT=NFLT+1
      FLT(NFLT)=FL(I)
      IVAR(K)=IFORM(5)
      GO TO 5
4     IVAR(K)=IFORM(6)
5     CONTINUE
      K=K+1
      IVAR(K)=IFORM(7)
      K=K+1
      IVAR(K)=IFORM(8)
      PRINT IVAR, (INT(I),I=1,NINT),(FLT(J),J=1,NFLT),(IA(L),L=1,NWORDS)
      RETURN
      END


      SUBROUTINE REFLC (IX,IY,IZ,ITX,NOP)
C
C     REFLC REFLECTS PARTIAL STRUCTURE ALONG X,Y, OR Z AXES OR ROTATES
C     STRUCTURE TO COMPLETE A SYMMETRIC STRUCTURE.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), X2(1), Y
     12(1), Z2(1)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG), (X2,SI), (Y2,ALP), (Z2,BET)
      NP=N
      MP=M
      IPSYM=0
      ITI=ITX
      IF (IX.LT.0) GO TO 19
      IF (NOP.EQ.0) RETURN
      IPSYM=1
      IF (IZ.EQ.0) GO TO 6
C
C     REFLECT ALONG Z AXIS
C
      IPSYM=2
      IF (N.EQ.0) GO TO 3
      DO 2 I=1,N
      NX=I+N
      E1=Z(I)
      E2=Z2(I)
      IF (ABS(E1)+ABS(E2).GT.1.E-5.AND.E1*E2.GE.-1.E-6) GO TO 1
      PRINT 24, I
      STOP
1     X(NX)=X(I)
      Y(NX)=Y(I)
      Z(NX)=-E1
      X2(NX)=X2(I)
      Y2(NX)=Y2(I)
      Z2(NX)=-E2
      ITAGI=ITAG(I)
      IF (ITAGI.EQ.0) ITAG(NX)=0
      IF (ITAGI.NE.0) ITAG(NX)=ITAGI+ITI
2     BI(NX)=BI(I)
      N=N*2
      ITI=ITI*2
3     IF (M.EQ.0) GO TO 6
      NXX=LD+1
      DO 5 I=1,M
      NXX=NXX-1
      NX=NXX-M
      IF (ABS(Z(NXX)).GT.1.E-10) GO TO 4
```

```
      PRINT 25, I
      STOP
4     X(NX)=X(NXX)
      Y(NX)=Y(NXX)
      Z(NX)=-Z(NXX)
      T1X(NX)=T1X(NXX)
      T1Y(NX)=T1Y(NXX)
      T1Z(NX)=-T1Z(NXX)
      T2X(NX)=T2X(NXX)
      T2Y(NX)=T2Y(NXX)
      T2Z(NX)=-T2Z(NXX)
      SALP(NX)=-SALP(NXX)
5     BI(NX)=BI(NXX)
      M=M*2
6     IF (IY.EQ.0) GO TO 12
C
C     REFLECT ALONG Y AXIS
C
      IF (N.EQ.0) GO TO 9
      DO 8 I=1,N
      NX=I+N
      E1=Y(I)
      E2=Y2(I)
      IF (ABS(E1)+ABS(E2).GT.1.E-5.AND.E1*E2.GE.-1.E-6) GO TO 7
      PRINT 24, I
      STOP
7     X(NX)=X(I)
      Y(NX)=-E1
      Z(NX)=Z(I)
      X2(NX)=X2(I)
      Y2(NX)=-E2
      Z2(NX)=Z2(I)
      ITAGI=ITAG(I)
      IF (ITAGI.EQ.0) ITAG(NX)=0
      IF (ITAGI.NE.0) ITAG(NX)=ITAGI+ITI
8     BI(NX)=BI(I)
      N=N*2
      ITI=ITI*2
9     IF (M.EQ.0) GO TO 12
      NXX=LD+1
      DO 11 I=1,M
      NXX=NXX-1
      NX=NXX-M
      IF (ABS(Y(NXX)).GT.1.E-10) GO TO 10
      PRINT 25, I
      STOP
10    X(NX)=X(NXX)
      Y(NX)=-Y(NXX)
      Z(NX)=Z(NXX)
      T1X(NX)=T1X(NXX)
      T1Y(NX)=-T1Y(NXX)
      T1Z(NX)=T1Z(NXX)
      T2X(NX)=T2X(NXX)
      T2Y(NX)=-T2Y(NXX)
      T2Z(NX)=T2Z(NXX)
      SALP(NX)=-SALP(NXX)
11    BI(NX)=BI(NXX)
      M=M*2
12    IF (IX.EQ.0) GO TO 18
C
C     REFLECT ALONG X AXIS
C
      IF (N.EQ.0) GO TO 15
      DO 14 I=1,N
      NX=I+N
      E1=X(I)
      E2=X2(I)
      IF (ABS(E1)+ABS(E2).GT.1.E-5.AND.E1*E2.GE.-1.E-6) GO TO 13
      PRINT 24, I
      STOP
13    X(NX)=-E1
      Y(NX)=Y(I)
      Z(NX)=Z(I)
      X2(NX)=-E2
      Y2(NX)=Y2(I)
      Z2(NX)=Z2(I)
      ITAGI=ITAG(I)
      IF (ITAGI.EQ.0) ITAG(NX)=0
      IF (ITAGI.NE.0) ITAG(NX)=ITAGI+ITI
14    BI(NX)=BI(I)
      N=N*2
15    IF (M.EQ.0) GO TO 18
      NXX=LD+1
      DO 17 I=1,M
      NXX=NXX-1
      NX=NXX-M
      IF (ABS(X(NXX)).GT.1.E-10) GO TO 16
      PRINT 25, I
      STOP
16    X(NX)=-X(NXX)
      Y(NX)=Y(NXX)
      Z(NX)=Z(NXX)
      T1X(NX)=-T1X(NXX)
      T1Y(NX)=T1Y(NXX)
      T1Z(NX)=T1Z(NXX)
      T2X(NX)=-T2X(NXX)
      T2Y(NX)=T2Y(NXX)
      T2Z(NX)=T2Z(NXX)
      SALP(NX)=-SALP(NXX)
```

```
17    BI(NX)=BI(NXX)
      M=M*2
18    RETURN
C
C     REPRODUCE STRUCTURE WITH ROTATION TO FORM CYLINDRICAL STRUCTURE
C
19    FNOP=NOP
      IPSYM=-1
      SAM=6.283185308/FNOP
      CS=COS(SAM)
      SS=SIN(SAM)
      IF (N.EQ.0) GO TO 21
      N=N*NOP
      NX=NP+1
      DO 20 I=NX,N
      K=I-NP
      XK=X(K)
      YK=Y(K)
      X(I)=XK*CS-YK*SS
      Y(I)=XK*SS+YK*CS
      Z(I)=Z(K)
      XK=X2(K)
      YK=Y2(K)
      X2(I)=XK*CS-YK*SS
      Y2(I)=XK*SS+YK*CS
      Z2(I)=Z2(K)
      ITAGI=ITAG(K)
      IF (ITAGI.EQ.0) ITAG(I)=0
      IF (ITAGI.NE.0) ITAG(I)=ITAGI+ITI
20    BI(I)=BI(K)
21    IF (M.EQ.0) GO TO 23
      M=M*NOP
      NX=MP+1
      K=LD+1
      DO 22 I=NX,M
      K=K-1
      J=K-MP
      XK=X(K)
      YK=Y(K)
      X(J)=XK*CS-YK*SS
      Y(J)=XK*SS+YK*CS
      Z(J)=Z(K)
      XK=T1X(K)
      YK=T1Y(K)
      T1X(J)=XK*CS-YK*SS
      T1Y(J)=XK*SS+YK*CS
      T1Z(J)=T1Z(K)
      XK=T2X(K)
      YK=T2Y(K)
      T2X(J)=XK*CS-YK*SS
      T2Y(J)=XK*SS+YK*CS
      T2Z(J)=T2Z(K)
      SALP(J)=SALP(K)
22    BI(J)=BI(K)
23    RETURN
C
24    FORMAT (29H GEOMETRY DATA ERROR--SEGMENT,I5,26H LIES IN PLANE OF S
     1YMMETRY)
25    FORMAT (27H GEOMETRY DATA ERROR--PATCH,I4,26H LIES IN PLANE OF SYM
     1METRY)
      END


      SUBROUTINE SOLVE (N,A,IP,B,NDIM)
C
C     SUBROUTINE TO SOLVE THE MATRIX EQUATION LU*X=B WHERE L IS A UNIT
C     LOWER TRIANGULAR MATRIX AND U IS AN UPPER TRIANGULAR MATRIX BOTH
C     OF WHICH ARE STORED IN A.  THE RHS VECTOR B IS INPUT AND THE
C     SOLUTION IS RETURNED THROUGH VECTOR B.    (MATRIX TRANSPOSED.
C
      DIMENSION A(NDIM,NDIM), IP(NDIM), B(NDIM)
      COMMON /SCRATM/ Y(1500)
      COMPLEX A,B,Y,SUM
      INTEGER PI
C
C     FORWARD SUBSTITUTION
C
      DO 3 I=1,N
      PI=IP(I)
      Y(I)=B(PI)
      B(PI)=B(I)
      IP1=I+1
      IF (IP1.GT.N) GO TO 2
      DO 1 J=IP1,N
      B(J)=B(J)-A(I,J)*Y(I)
1     CONTINUE
2     CONTINUE
3     CONTINUE
C
C     BACKWARD SUBSTITUTION
C
      DO 6 K=1,N
      I=N-K+1
      SUM=(0.,0.)
      IP1=I+1
      IF (IP1.GT.N) GO TO 5
      DO 4 J=IP1,N
      SUM=SUM+A(J,I)*B(J)
```

```
4     CONTINUE
5     CONTINUE
      B(I)=(Y(I)-SUM)/A(I,I)
6     CONTINUE
      RETURN
      END

      SUBROUTINE SOLVES (NOP,A,IP,B,NROW,NCOL,IX,NP,N,MP,M)
C
C     SUBROUTINE SOLVES, FOR SYMMETRIC STRUCTURES, HANDLES THE
C     TRANSFORMATION OF THE RIGHT HAND SIDE VECTOR AND SOLUTION OF THE
C     MATRIX EQ.
C
      COMMON /SMAT/ S(10,10)
      DIMENSION A(NROW*NCOL), IP(1), IX(1), B(1)
      COMMON /SCRATM/ Y(1500)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMPLEX A,B,Y,SUM,S
      NPEQ=NP+2*MP
      NEQ=N+2*M
      IF (NOP.EQ.1) GO TO 11
      IF (N.EQ.0.OR.M.EQ.0) GO TO 6
      DO 1 I=1,NEQ
1     Y(I)=B(I)
      KK=2*MP
      IA=NP
      IB=N
      J=NP
      DO 5 K=1,NOP
      IF (K.EQ.1) GO TO 3
      DO 2 I=1,NP
      IA=IA+1
      J=J+1
2     B(J)=Y(IA)
      IF (K.EQ.NOP) GO TO 5
3     DO 4 I=1,KK
      IB=IB+1
      J=J+1
4     B(J)=Y(IB)
5     CONTINUE
6     FNOP=NOP
      FNORM=1./FNOP
C
C     TRANSFORM MATRIX EQ. RHS VECTOR ACCORDING TO SYMMETRY MODES
C
      DO 10 I=1,NPEQ
      DO 7 K=1,NOP
      IA=I+(K-1)*NPEQ
7     Y(K)=B(IA)
      SUM=Y(1)
      DO 8 K=2,NOP
8     SUM=SUM+Y(K)
      B(I)=SUM*FNORM
      DO 10 K=2,NOP
      IA=I+(K-1)*NPEQ
      SUM=Y(1)
      DO 9 J=2,NOP
9     SUM=SUM+Y(J)*CONJG(S(K,J))
10    B(IA)=SUM*FNORM
11    IF (ICASE.LT.3) GO TO 12
      REWIND 15
      REWIND 16
C
C     SOLVE EACH MODE EQUATION
C
12    DO 16 KK=1,NOP
      IA=(KK-1)*NPEQ+1
      IB=IA
      IF (ICASE.NE.4) GO TO 14
      DO 13 I=1,NPEQ
13    READ (15) (A(J,I),J=1,NPEQ)
      IB=1
14    IF (ICASE.EQ.3.OR.ICASE.EQ.5) GO TO 15
      CALL SOLVE (NPEQ,A(IB,1),IP(IA),B(IA),NROW)
      GO TO 16
15    CALL LTSOLV (A,NPEQ,NCOL,IX(IA),B(IA))
16    CONTINUE
      IF (NOP.EQ.1) RETURN
C
C     INVERSE TRANSFORM THE MODE SOLUTIONS
C
      DO 20 I=1,NPEQ
      DO 17 K=1,NOP
      IA=I+(K-1)*NPEQ
17    Y(K)=B(IA)
      SUM=Y(1)
      DO 18 K=2,NOP
18    SUM=SUM+Y(K)
      B(I)=SUM
      DO 20 K=2,NOP
      IA=I+(K-1)*NPEQ
      SUM=Y(1)
      DO 19 J=2,NOP
19    SUM=SUM+Y(J)*S(K,J)
20    B(IA)=SUM
      IF (N.EQ.0.OR.M.EQ.0) RETURN
      DO 21 I=1,NEQ
```

```
21    Y(I)=B(I)
      KK=2*NP
      IA=NP
      IB=N
      J=NP
      DO 25 K=1,NOP
      IF (K.EQ.1) GO TO 23
      DO 22 I=1,NP
      IA=IA+1
      J=J+1
22    B(IA)=Y(J)
      IF (K.EQ.NOP) GO TO 25
23    DO 24 I=1,KK
      IB=IB+1
      J=J+1
24    B(IB)=Y(J)
25    CONTINUE
      RETURN
      END


      SUBROUTINE SUB1 (NP,A,NOP,IP,N)
C
C     SUB1 IS USED FOR THE OUT OF CORE SYMMETRY CASE WHEN THE NP*NP
C     SUBMATRIX FITS IN CORE. IT CALLS FACTR NOP TIMES
C
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      DIMENSION A(NP,NP), IP(N)
      COMPLEX A
      NC1=IC2+1
      IF (IC2.EQ.0) GO TO 3
      IF (IC2.EQ.NOP) GO TO 7
      DO 2 I=1,IC2
      DO 1 K=1,NP
      READ (13) (A(J,K),J=1,NP)
      READ (15) (A(J,K),J=1,NP)
1     CONTINUE
2     CONTINUE
3     CONTINUE
      DO 6 KK=NC1,NOP
      DO 4 I=1,NP
4     READ (13) (A(J,I),J=1,NP)
      KA=(KK-1)*NP+1
      CALL FACTR (NP,A,IP(KA),NP)
      DO 5 I=1,NP
5     WRITE (15) (A(J,I),J=1,NP)
      IC2=KK
      CALL CHKPRT
6     CONTINUE
      REWIND 15
7     RETURN
      END


      SUBROUTINE SUB2 (A,NP,NCOLS,NOP,IP,IX,N)
C
C     SUB2 IS USED IN THE OUT OF CORE SYMMETRY CASE WHEN THE NP*NP
C     SUBMATRIX DOES NOT FIT IN CORE.  IT CALLS FACIO AND LUNSCR NOP
C     TIMES.
C
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      DIMENSION A(NP,NCOLS), IP(NP), IX(N)
      COMPLEX A
      NBLOKS=NBLSYM
      NPBLK=NPSYM
      NLAST=NLSYM
      I2=2*NPBLK*NP
      NC1=IC2+1
      IF (IC2.EQ.0) GO TO 2
      IF (IC2.EQ.NOP) GO TO 8
      DO 1 I=1,IC2
      CALL BLCKIN (11,1,I2,NBLOKS,190)
      CALL BLCKIN (15,1,I2,NBLOKS,191)
      CALL BLCKIN (16,1,I2,NBLOKS,192)
1     CONTINUE
      GO TO 6
2     CONTINUE
      IF (IC1.EQ.-1) GO TO 6
      DO 5 KK=1,NOP
      J2=NPBLK
      DO 4 L=1,NBLOKS
      IF (L.EQ.NBLOKS) J2=NLAST
      DO 3 J=1,J2
3     READ (13) (A(I,J),I=1,NP)
      CALL BLCKOT (11,1,I2,1,193)
4     CONTINUE
5     CONTINUE
      IC1=-1
      REWIND 11
6     DO 7 KK=NC1,NOP
      KA=(KK-1)*NP+1
      CALL FACIO (A,NP,NCOLS,IP)
      CALL LUNSCR (A,NP,NCOLS,IX(KA),IP)
      IC2=KK
      CALL CHKPRT
7     CONTINUE
```

```
      REWIND 15
      REWIND 16
      IC3=-1
    8 RETURN
      END


      SUBROUTINE TEST (F1R,F2R,TR,F1I,F2I,TI)
      IF (ABS(F2R).GT.ABS(F2I)) GO TO 1
      DEN=F2I
      GO TO 2
    1 DEN=F2R
    2 IF (ABS(DEN).LT.1.E-37) GO TO 3
      TR=ABS((F1R-F2R)/DEN)
      TI=ABS((F1I-F2I)/DEN)
      RETURN
    3 TR=0.
      TI=0.
      RETURN
      END


      SUBROUTINE TRIO (J,JCO1,JCO2,DIL,DIK)
C
C     SUBROUTINE TRIO DETERMINES WHICH SEGMENTS ARE CONNECTED TO SEGMENT
C     J.  SUBROUTINE JUNC IS CALLED TO FILL COMMON/JUNK/ FOR MULTIPLE
C     JUNCTIONS.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /JUNK/ NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
      S=SI(J)
      JCO1=ICON1(J)
      JCO2=ICON2(J)
      IF (JCO1) 1,2,3
    1 CALL JUNC (J,JCO1,NCOX,JOX,NCIX,JIX,DIL)
      GO TO 5
    2 DIL=S/2.0
      GO TO 5
    3 IF (JCO1.LT.10000) GO TO 4
      DIL=S
      GO TO 5
    4 DIL=(SI(JCO1)+S)/2.0
    5 IF (JCO2) 6,7,8
    6 CALL JUNC (J,JCO2,NCOZ,JOZ,NCIZ,JIZ,DIK)
      GO TO 10
    7 DIK=S/2.0
      GO TO 10
    8 IF (JCO2.LT.10000) GO TO 9
      DIK=S
      GO TO 10
    9 DIK=(SI(JCO2)+S)/2.0
   10 CONTINUE
      RETURN
      END


      SUBROUTINE UDOTES (I1,I2,CM,NROW,NCOL)
C
C     COMPUTES E FIELD ALONG WIRES DUE TO SURFACE CURRENTS AND FILLS
C     APPROPRIATE MATRIX ELEMENTS
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      COMMON /ANGL/ SALP(1000)
      COMMON /GND/ ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,IPERF
      COMMON /DATI/ T1XI,T1YI,T1ZI,T2XI,T2YI,T2ZI,XI,YI,ZI
      DIMENSION CAB(1), SAB(1), CM(NROW,NCOL)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), XS(1), Y
     1S(1), ZS(1), S(1), EMFL(9)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG), (XS,X), (YS,Y), (ZS,Z), (S,BI)
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1))
      COMPLEX E1X,E1Y,E1Z,E2X,E2Y,E2Z,CM,EMEL
      COMPLEX ZRATI,ZRATI2
      IF (I1.GT.NP) RETURN
      IEND=I2
      IF (I2.GT.NP) IEND=NP
      K=0
      NOP=M/MP
      ICGO=1
C     OBSERVATION LOOP
      DO 6 I=I1,IEND
      K=K+1
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)
      CABI=CAB(I)
      SABI=SAB(I)
      SALPI=SALP(I)
      JS=LD+1
      IPCH=0
      IF (ICON1(I).LT.10000) GO TO 1
      IPCH=ICON1(I)-10000
```

```
      IPCH=JS-IPCH
      FSIGN=-1.
 1    IF (ICON2(I).LT.10000) GO TO 2
      IPCH=ICON2(I)-10000
      IPCH=JS-IPCH
      FSIGN=1.
C     SOURCE LOOPS
 2    DO 6 L=1,NOP
      JL=L*(NP+2*MP)-2*MP
      DO 6 J=1,MP
      JS=JS-1
      JL=JL+2
      T1XI=T1X(JS)
      T1YI=T1Y(JS)
      T2XI=T2X(JS)
      T2YI=T2Y(JS)
      CM(JL-1,K)=(0.,0.)
      CM(JL,K)=(0.,0.)
      RFL=-1.
C     GROUND LOOP
      DO 6 IP=1,KSYMP
      RFL=-RFL
      T1ZI=T1Z(JS)*RFL
      T2ZI=T2Z(JS)*RFL
      IF (IPCH.NE.JS.AND.ICGO.EQ.1) GO TO 4
      IF (IP.EQ.2) GO TO 4
      IF (ICGO.GT.1) GO TO 3
      CALL PCINT (XS(JS),YS(JS),ZS(JS),S(JS),CABI,SABI,SALPI,EMEL)
      IL=I+2*MP*(L-1)
      CM(IL,K)=CM(IL,K)+EMEL(9)*FSIGN
 3    CM(JL-1,K)=EMEL(ICGO)
      CM(JL,K)=EMEL(ICGO+4)
      ICGO=ICGO+1
      IF (ICGO.EQ.5) ICGO=1
      GO TO 5
 4    ZSEP=ZI-ZS(JS)*RFL
      CALL UNERE (XI-XS(JS),YI-YS(JS),ZSEP,S(JS),E1X,E1Y,E1Z,E2X,E2Y,E2Z
     1)
      CM(JL-1,K)=(E1X*CABI+E1Y*SABI+E1Z*SALPI)*RFL+CM(JL-1,K)
      CM(JL,K)=(E2X*CABI+E2Y*SABI+E2Z*SALPI)*RFL+CM(JL,K)
 5    CONTINUE
 6    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UNCAT
C
C     UNCAT READS FILE 17 AND SETS UP FILES 11-16 FOR RESTARTING PROGRAM
C
      COMMON CM(4000)
      COMMON /RESTRT/ IC1,IC2,IC3,NRES,NPRES,IBLCK,IDUMP,TMDUM,EXTIM
      COMMON /SAVE/ IX(1500),IP(1500)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      DIMENSION IT(7)
      COMPLEX CM
      LOGICAL ENF
      EQUIVALENCE (N,NRES), (NP,NPRES), (I2,IBLCK)
      DATA (IT(I),I=1,7)/11,12,13,14,15,16,17/
      DO 1 I=1,7
      NUNIT=IT(I)
 1    REWIND NUNIT
      READ (17) IC1,IC2,IC3,N,NP,I2,ICASE
      READ (17) IX(1)
      IF (ENF(17)) GO TO 2
 2    CONTINUE
      ISIX=6
      IF (ICASE.EQ.4) ISIX=1
      DO 5 I=1,ISIX
      NUNIT=IT(I)
      NEOF=777
      IFLCNT=0
 3    CALL BLCKIN (17,1,I2,1,NEOF)
      IF (NEOF.EQ.0) GO TO 4
      CALL BLCKOT (NUNIT,1,I2,1,1)
      IFLCNT=IFLCNT+1
      GO TO 3
 4    END FILE NUNIT
      PRINT 12, IFLCNT,NUNIT
 5    CONTINUE
      IF (ICASE.NE.4) GO TO 10
      IFLCNT=0
 6    READ (17) (CM(I),I=1,NP)
      IF (ENF(17)) GO TO 7
      WRITE (13) (CM(I),I=1,NP)
      IFLCNT=IFLCNT+1
      GO TO 6
 7    END FILE 13
      PRINT 12, IFLCNT,IT(3)
      IFLCNT=0
 8    READ (17) (CM(I),I=1,NP)
      IF (ENF(17)) GO TO 9
      WRITE (15) (CM(I),I=1,NP)
      IFLCNT=IFLCNT+1
      GO TO 8
 9    END FILE 15
      PRINT 12, IFLCNT,IT(5)
      IFLCNT=0
```

```
C
10    CONTINUE
      READ (17) (IX(I),I=1,N)
      READ (17) (IP(I),I=1,N)
      DO 11 I=1,7
      NUNIT=IT(I)
11    REWIND NUNIT
      RETURN
C
12    FORMAT (/,I5,28H  FILES RETRIEVED FROM UNIT ,I5/)
      END
```

```
      SUBROUTINE UNERE (RX,RY,RZ,S,E1X,E1Y,E1Z,E2X,E2Y,E2Z)
C
C     CALCULATES THE ELECTRIC FIELD DUE TO UNIT CURRENT IN THE T1 AND T2
C     DIRECTIONS ON A PATCH
C
      COMMON /DATI/ T1XI,T1YI,T1ZI,T2XI,T2YI,T2ZI,XI,YI,ZI
      COMPLEX ER,Q1,Q2,E1X,E1Y,E1Z,E2X,E2Y,E2Z
      DATA TPI,CONST/6.2831853l,4.7713412/
C     CONST=ETA/(8.*PI**2)
      R2=RX*RX+RY*RY+RZ*RZ
      R=SQRT(R2)
      T1=-TPI*R
      T2=T1*T1
      RT=R2*R
      ER=CMPLX(SIN(T1),-COS(T1))*(CONST*S)
      Q1=CMPLX(T2-1.,T1)*ER/RT
      Q2=CMPLX(3.-T2,-3.*T1)*ER/(RT*R2)
      ER=Q2*(T1XI*RX+T1YI*RY+T1ZI*RZ)
      E1X=Q1*T1XI+ER*RX
      E1Y=Q1*T1YI+ER*RY
      E1Z=Q1*T1ZI+ER*RZ
      ER=Q2*(T2XI*RX+T2YI*RY+T2ZI*RZ)
      E2X=Q1*T2XI+ER*RX
      E2Y=Q1*T2YI+ER*RY
      E2Z=Q1*T2ZI+ER*RZ
      RETURN
      END
```

```
      SUBROUTINE WIRE (XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,NS,ITG)
C
C     SUBROUTINE WIRE GENERATES SEGMENT GEOMETRY DATA FOR A STRAIGHT
C     WIRE OF NS SEGMENTS.
C
      COMMON /DATA/ LD,N,NP,M,MP,X(1000),Y(1000),Z(1000),SI(1000),BI(100
     10),ALP(1000),BET(1000),ICON1(1000),ICON2(1000),ITAG(1000),WLAM,IPS
     2YM
      DIMENSION X2(1), Y2(1), Z2(1)
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      IST=N+1
      N=N+NS
      NP=N
      MP=M
      IPSYM=0
      IF (NS.LT.1) RETURN
      FNS=NS
      XD=(XW2-XW1)/FNS
      YD=(YW2-YW1)/FNS
      ZD=(ZW2-ZW1)/FNS
      XS1=XW1
      YS1=YW1
      ZS1=ZW1
      DO 1 I=IST,N
      ITAG(I)=ITG
      XS2=XS1+XD
      YS2=YS1+YD
      ZS2=ZS1+ZD
      X(I)=XS1
      Y(I)=YS1
      Z(I)=ZS1
      X2(I)=XS2
      Y2(I)=YS2
      Z2(I)=ZS2
      BI(I)=RAD
      XS1=XS2
      YS1=YS2
1     ZS1=ZS2
      RETURN
      END
```

```
      COMPLEX FUNCTION ZINT(SIGL,ROLAM)
C
C     ZINT COMPUTES THE INTERNAL IMPEDANCE OF A CIRCULAR WIRE
C
      COMPLEX TH,PH,F,G,FJ,CN,BR1,BR2
      DATA PI,POT,TP,TPCMU,CMOTP,FJ,CN/3.1415926,1.5707963,6.2831853,2.3
     168705E+3,60.00,(0.,1.),(.70710678,.70710678)/
      TH(D)=(((((6.E-7,1.9E-6)*D+(-3.4E-6,5.1E-6))*D+(-2.52E-5,0.))*D+(
     1-9.06E-5,-9.01E-5))*D+(0.,-9.765E-4))*D+(.0110486,-.0110485))*D+(0
     2.,-.3926991)
      PH(D)=(((((1.6E-6,-3.2E-6)*D+(1.17E-5,-2.4E-6))*D+(3.46E-5,3.38E-
     15))*D+(5.E-7,2.452E-4))*D+(-1.3813E-3,1.3811E-3))*D+(-6.25001E-2,-
```

```
21.E-7))*D+(.7071068,.7071068)
 F(D)=SQRT(POT/D)*CEXP(-CN*D*TH(-8./X))
 G(D)=1./SQRT(TP*D)*CEXP(CN*D*TH(8./X))
 X=SQRT(TPCMU*SIGL)*ROLAM
 IF (X.GT.110.) GO TO 2
 IF (X.GT.8.) GO TO 1
 V=X/8.
 V=V*V
 S=V*V
 BER=((((((-9.01E-6*S+1.22552E-3)*S-.08349609)*S+2.64191491)*S-32.36
13456)*S+113.77778)*S-64.)*S+1.
 BEI=((((((1.1346E-4*S-.01103667)*S+.52185615)*S-10.56765801)*S+72.81
17777)*S-113.77778)*S+16.)*V
 BR1=CMPLX(BER,BEI)
 BER=(((((((-3.94E-6*S+4.5957E-4)*S-.02609253)*S+.66047849)*S-6.068
11481)*S+14.222222)*S-4.)*V)*X
 BEI=((((((4.609E-5*S-3.79386E-3)*S+.14677204)*S-2.31167514)*S+11.37
17778)*S-10.666667)*S+.5)*X
 BR2=CMPLX(BER,BEI)
 BR1=BR1/BR2
 GO TO 3
1 BR1=G(X)+FJ*F(X)/PI
 BR2=G(X)*PH(8./X)-FJ*F(X)*PH(-8./X)/PI
 BR1=BR1/BR2
 GO TO 3
2 BR1=(.70710678,-.70710678)
3 ZINT=FJ*SQRT(CMOTP/SIGL)*BR1/ROLAM
 RETURN
 END
```

REFERENCES

1. MBAssociates,"Antenna Modeling Program - Composite User's Manual," MBAssociates Report No. MB-R-74/46, 1974. A-005 673

2. MBAssociates,"Antenna Modeling Program - Engineering Manual," MBAssociates Report No. MB-R-74/62, 1974.

3. MBAssociates,"Antenna Modeling Program - Systems Manual," MBAssociates Report No. MB-R-74/62, 1974.

4. Poggio, A. J. and E. K. Miller,"Integral Equation Solutions of Three-Dimensional Scattering Problems," MBAssociates Technical Memo No. MB-TM-70/20, July 1970.

5. Albertsen, N. C., J. E. Hansen, N. E. Jensen,"Computation of Spacecraft Antenna Radiation Patterns," The Technical University of Denmark, Lyngby, Denmark, June 1972.

6. Harrington, R. F., Time-Harmonic Electromagnetic Fields, McGraw-Hill (1961), p. 79.